VSB — TECHNICAL UNIVERSITY OF OSTRAVA
FACULTY OF ECONOMICS

DEPARTMENT OF MANAGEMENT

Optimalizace portfolia s splikací v Matlabu
Portfolio Optimization with Application in Matlab

Student: Yin Xu
Supervisor of the diploma thesis: Ing. Aleš Kresta, Ph.D

.

Ostrava 2017

VŠB - Technical University of Ostrava
Faculty of Economics
Department of Finance

Diploma Thesis Assignment

Student:	**Bc. Yin Xu**
Study Programme:	N6202 Economic Policy and Administration
Study Branch:	6202T010 Finance
Title:	Portfolio Optimization with Application in Matlab Optimalizace portfolia s aplikací v Matlabu
The thesis language:	English

Description:

1. Introduction
2. Description of Matlab
3. Description of Portfolio Optimization Models
4. Portfolio Optimization Backtesting in Matlab
5. Conclusion

Bibliography
List of Abbreviations
Declaration of Utilisation of Results from the Diploma Thesis
List of Annexes
Annexes

References:

KIENITZ, Jörg and Daniel WETTERAU. *Financial modelling: theory, implementation and practice with MATLAB source*. Chichester: Wiley, 2012. ISBN 978-0-470-74489-5.
KRESTA, Aleš. *Financial Engineering in Matlab: Selected Approaches and Algorithms*. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3702-4.
ZMEŠKAL, Z., D. DLUHOŠOVÁ and T. TICHÝ. *Financial Models*. Ostrava: VŠB-TU Ostrava, 2004. ISBN 80-248-0754-8.

Extent and terms of a thesis are specified in directions for its elaboration that are opened to the public on the web sites of the faculty.

Supervisor: **Ing. Aleš Kresta, Ph.D.**

Date of issue: 18.11.2016
Date of submission: 21.04.2017

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Ekonomická fakulta 1 OSTRAVA

Ing. Iveta Ratmanová, Ph.D.
Head of Department

prof. Dr. Ing. Zdeněk Zmeškal
Dean

The Declaration

"Herewith I declare that I elaborated the entire thesis, including all annexes, independently."

Ostrava dated 18.04.2017

Yin Xu 徐寅

YIN XU

Contents

1. Introduction

Portfolio optimization is a procedure to select a variety of stocks or other assets based on investors' need and create a portfolio according to the demand of the investors. The nature of portfolio optimization is to choose between risk and returns. Investors want to maximize the return with less risk. This theory was firstly proposed by Markowitz (1952), who proposed that if the investors need to choose between two portfolios with the same return, all investors will choose the less risky one. This phenomenon implies that investors need to take high risks for high returns. Similarly, investors usually hold various portfolios to diversify the risk and improve the utility.

The goal of the thesis is verify and compare the out-of-sample performance of following strategies: naive, Markowitz and conditional value at risk. For calculation, we choose 43 stocks that are components of Hang Seng Index from Hong Kong Stock Exchange and apply the data from 2006 to 2016. The calculation is mainly done using Matlab.

The thesis is divided into five parts. The first chapter is the introduction. The second chapter provides basic description of Matlab. In this chapter we firstly introduce the basic history and development of Matlab. Secondly some basic environment information is introduced. Thirdly, there are some basic structures of Matlab, which include the sequential structure and selective structure. Then there is program error and debugger, and graphing in Matlab.

The third chapter is the introduction of methodology for the thesis. In this chapter we introduce the description of portfolio optimization models. Firstly the risk measures, including standard deviation, value at risk and conditional value at risk. Then we introduce the naive strategy, mean-variance model, mean-conditional value at risk model and backtesting procedure of these models.

The fourth chapter is the application part. We apply naive strategy, mean-variance model and mean-CVaR model for chosen stocks in Matlab software and then do the backtesting and comparison of results.

2.Description of Matlab

Matlab, short for Matrix Laboratory, is a computation tool based on matrix manipulations. Matlab is an integration of high performance numerical computing and data visualization and programming. Mathematical functions are provided in Matlab, for example linear algebra, Fourier analysis, statistics, optimization and numerical integration. In this chapter the basic description of Matlab is introduced. The methodology is based on the Moore (2011), Zhang (2014) and Beucher and Weeks (2007).

2.1 Basic Introduction of Matlab

Matlab was designed by Cleve Moler from the University of New Mexico for matrix calculation. After years of development, Matlab has been applied in various kinds of works.. Now Matlab is developed by MathWorks using language C. As the version update of Matlab, the functions of Matlab have improvements, including expand in Toolbox and other module. In numerical processing aspects, it increased mathematic functions and renew the function and calculation of some functions. In external interface aspects, Matlab added the Java interface. It also added the Simulink toolbox to get three-dimensional dynamic function. Now language C, C++, C#, Java, Fortran and Python can be applied in Matlab, make it more flexible to use.

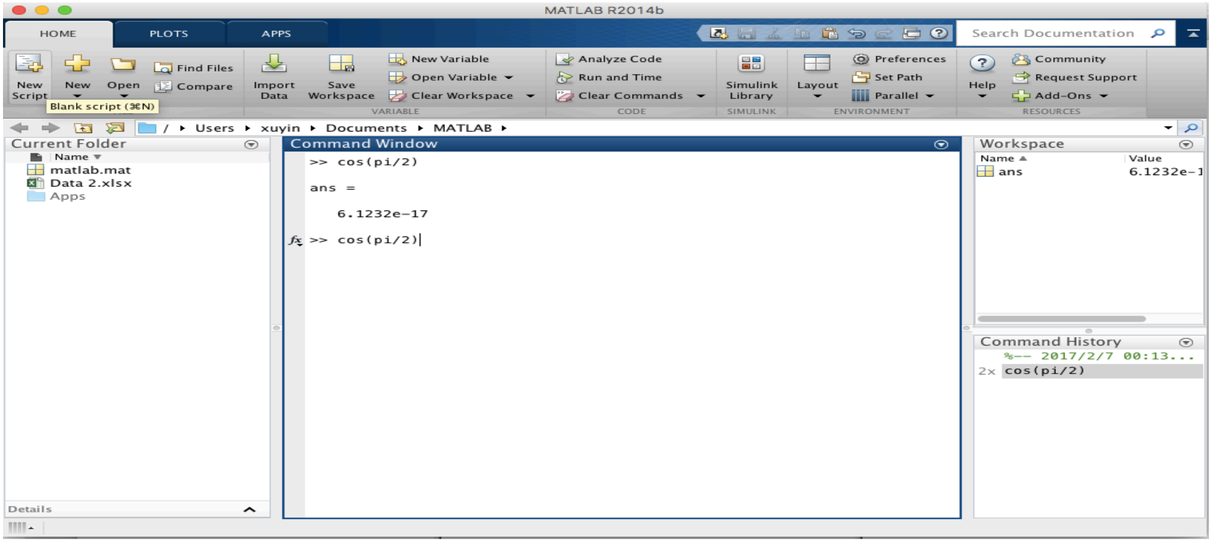
Matlab products is formed by different modules with different functions. The modules include Matlab, Matlab Toolboxes, MATLAB Compiler, Simulink, Simulink Blockset, Real-TimeWorkshop(RTW), Stateflow and Stateflow Coder. Among them, Matlab is the basis of Matlab product system. It provides basic mathematical algorithms, such as matrix calculation, and numerical analysis. MATLAB is the integrates 2D and 3D graphics capabilities that can achieve numerical visualization and provides an interactive high-level programming language - M language. M language can be used in writing script and function to get the user's own algorithm.

2.2 Environment of Matlab

The development environment of Matlab is the basic and core part of Matlab language. All function of Matlab language should be applied in Matlab development environment. The Simulink, Toolbox and other functions also should work under Matlab. To get better understanding of Matlab, the development environment of Matlab is the key factor.

When we open the Matlab, the desktop shows the default layout. Figure 2.1 shows the desktop of Matlab. The desktop of Matlab includes Command Window, Current Folder, Command History, Workspace and Home menu. Also the Launch Pad, M editor and Array Editor can be added to desktop from the layout option.

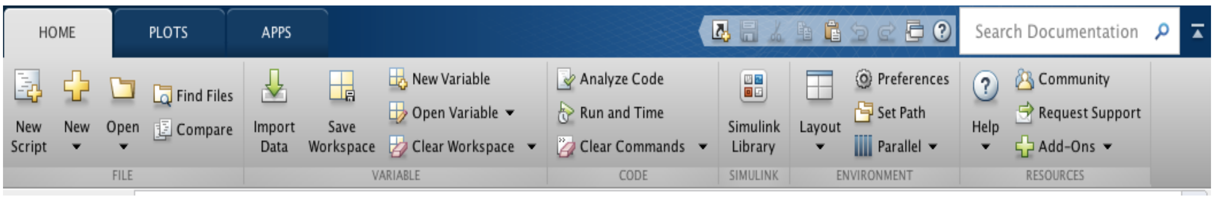
Figure 2.1 Desktop of Matlab



2.2.1 Home Menu of Matlab

Home menu is in the top of Matlab desktop, which is as figure 2.2 shows. The menu includes file, variable, code, Simulink, environment and resources menu.

Figure 2.2 Home menu of Matlab



In the file menu, there are several option. The option “new” can be used for opening a new file. The option “open’ is to open existing file, includes M file, fig file, mat file, mdl file and cdr file. In the variable menu, there’s import data, save workspace, new variable, open variable and clear workspace. The Code menu is for analysis of potential errors and make improvement of codes. For example, a common message indicates that a variable foo might be unused.

Figure 2.3 List of Simulink library

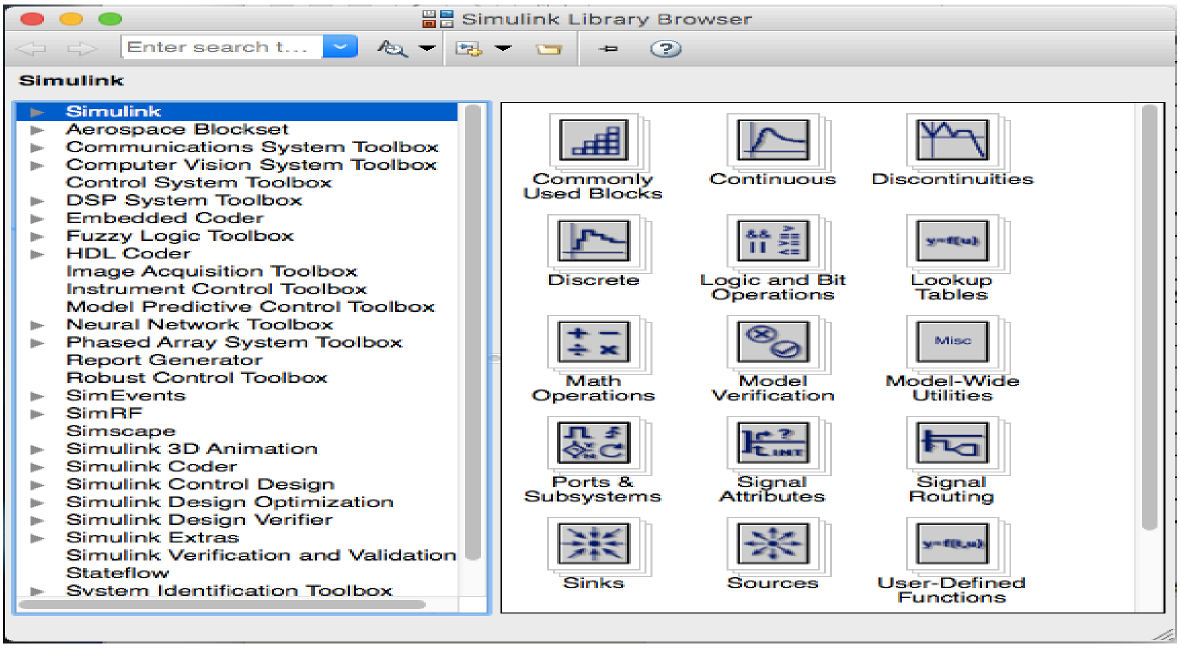


Figure 2.3 shows the list of Simulink library in Matlab. Simulink is one of Matlab's main toolbox. It provides a interactive dynamic system for modeling, simulating and analyzing graphical environment. Simulink is designed for controlling system, signal processing and communication system to do the functioning, simulation and analysis. It can solve linear and nonlinear systems, discrete, continuous and hybrid systems. It can solve both single task and multi-task discrete event system.

2.2.2 Command Window of Matlab

The command window is located in the center of the Matlab desktop. The command window can help saving the values calculated. However, for saving the command sequence that is need to generate the values. Also M-file is needed, which we will introduce later. The command window can be also undocked to a floating window for convenience. To floating the window, user need to click the  bottom on the command window.

Figure 2.4 Example of floating command window

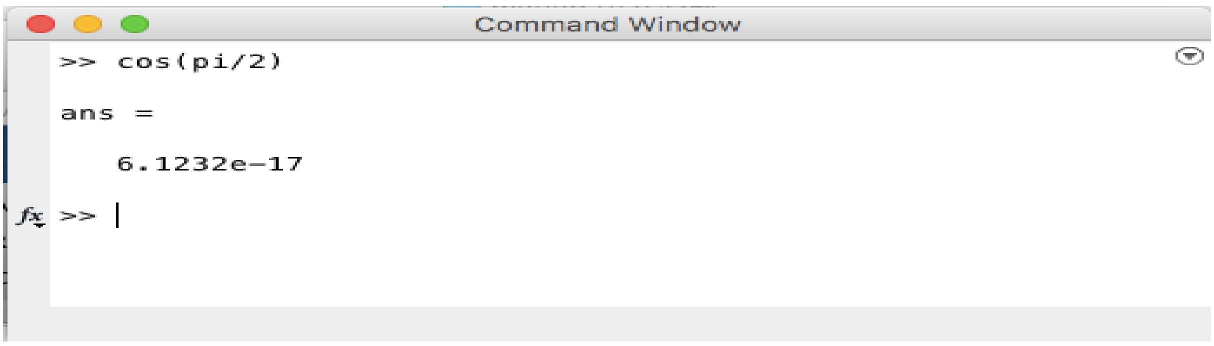


Figure 2.4 shows an example of a calculation in floating command window. The meaning of special characters is showed in Table 2.1. Applying these characters in Matlab can let it discriminate different functions. For example, when we write functions in Matlab, we usually use ";" character after finish one command sentence to separate rows and suppresses result. By using [*a b c*] we can set data *a, b* and *c* as a matrix for calculation. When using brackets it has to be balanced. Using "%% " we can divide the commands into different section, which can make it connivance to run the commands by section.

Table 2.1 Meaning of special characters in Matlab

Characters	Function
[ ]	forms a matrices of vector
( )	group operation, identify specific components
,	separates between subscripts or matrix components
;	separates rows in a matrix, suppresses result
:	generate matrices, indicates all rows or all columns
%	shows a comment in an M- le
%%	cell divider
+	add scalar and array
-	minus scalar and array
*	multiply scalar or matrix algebra
.*	multiply array
/	divided scalar or matrix algebra
./	divided array
^	exponential of scalar or matrix algebra
.^	exponential of array

Source: Moore (2010)

In the calculation ">>" is a command line prompt that indicates Matlab is in the ready status. After typing the formula and using the "enter" key, the command window will show the result and remain in ready status. The line "ans" is the abbreviation of answer. From figure 2.4 we can see the. It can make the result showed differential of the formula. Answer can also be used as a variable. If the command needed again, users only need to press the ↑ key. Table 2.2 is the keyboard shortcuts that can be used in the *input* area of MATLAB

Table 2.2 Keyboard shortcuts that can be used in the *input* area of MATLAB

Key	Function
↑ , Ctrl + P	Retrieves previous line
↓ , Ctrl + N	Retrieves following entry
← , Ctrl + B	Move cursor to the left for a character
→, Ctrl + F	Move cursor to the right for a character
Ctrl+←	Move cursor to the left for a word
Ctrl+→	Move cursor to the right for a word
Home, Ctrl +A	Move cursor to the beginning of the line
End, Ctrl + E	Move cursor to the end of the line
Esc	Clear the command line
Delete ,Ctrl + D	Deletes character that cursor indicated
Bank Space	Deletes character of the left of cursor
CTRL + K	Deletes all of the current line
Ctrl +Home	Move cursor to beginning of command window
Ctrl+ End	Move cursor to end of command window

Source: Moore (2010)

In Matlab, the keyboard short cut is very useful. It can help users easily find the previous line and calculations or move cursors freely. The shortcuts should be used in *input* area. Users can use 'Esc' to clear the command line.

Table 2.3 Format command of Matlab

Command	Function	Example
Short format	Number is with 4 decimal	2.7183
Long format	Number with 16 decimal places	2.718281828459046
Format long e	Number is with 16 decimal places using the power of 10	2.718281828459046e+00
Format short e	Number with four decimal places with the power of 10	2.7183e+00
Format long g	Results are showed in optimal of long format	2.71828182845905
Format short g	Results are showed in optimal of short format	2.7183
Bank format	Number is with 2 decimal places	2.72
Format rat	Number in rational fraction approximation	1457/536
Format hex	Number is in hexadecimal system	4005bf0a8b14576a

Source: Moore (2010)

Table 2.3 shows the format command in Matlab. In different command will let Matlab display different outplays of format. In the command window, the result of numerical calculation is in a fixed format. Matlab default digital display format is short format. Under short format the number is showed in a fixed format with number accuracy of four digits. For numbers greater than 1000, the number is expressed by scientific notation. To set the format of the data, users need to apply the format command. The example is the Euler's number.

2.2.3 Command History

In the default desktop of Matlab, the command history window is in the lower right comer. Similar as the command window, the command history window can be undocked. The command history window shows a log of statement that user ran in current and historic Matlab sessions. The command history presents the date and time of historic sessions and the commands written. Commands that run as a group are showed by the brackets in the left. The color mark indicates an error in the history. The history commands are not only clearly recorded in the Command History window, but can also be executed again. They can be copied into the MATLAB command window and be created M file directly from the instructions of these record. These functions are available through the shortcut menu command history window.

Diary Command is also commonly used in Matlab. Using diary command to create a diary to save the command and text typed in command window. The output of diary is in form of ASCII file that can be searched in, printed in most reports and other documents. Matlab creates a file named diary in the current folder on default.

Table 2.4 Function of Diary command

Command	Function
diary	Switch between diary on and off mode
diary on	Turn on diary mode using the current filename
diary off	Stop diary mode
diary(’filename’)	Create diary and name as ’filename’

Source: Moore (2010)

2.2.4 Current Folder

Current folder in Matlab is a browser for current path. Loading file and executing command in Matlab starts from the current folder. In the default desktop, the current folder window is in the left side. Under the full version of Matlab, users can also check the M file and MAT file. If there's help-document in M file, the information will be listed in the window. It also contains the variable in MAT file. The main goal of current folder is to help organizing the M files in current path. With the help of the current folder user can load and edit the files needed.

Under the Preference menu, there's setting for the current folder. The most important item is History. The History item helps setting the number of path showed in current folder. The default number is 20. Current folder can save the access record that can let users find the path quickly. When the path changed or user don't want to keep this information, the user can choose "Clear History" to delete the information.

2.2.5 Workspace

The Workspace of Matlab contains all variables from data. It reflects the name, bytes, size and type of the variables. Different icons represent the matrix, character array, unit and structure of variables. The variables in workspace will be deleted after exit Matlab. Save functions can be applied to save the variable to a MAT-file. Table 2.5 shows some basic functions in workspace.

Table 2.5 Basic Functions for workspace

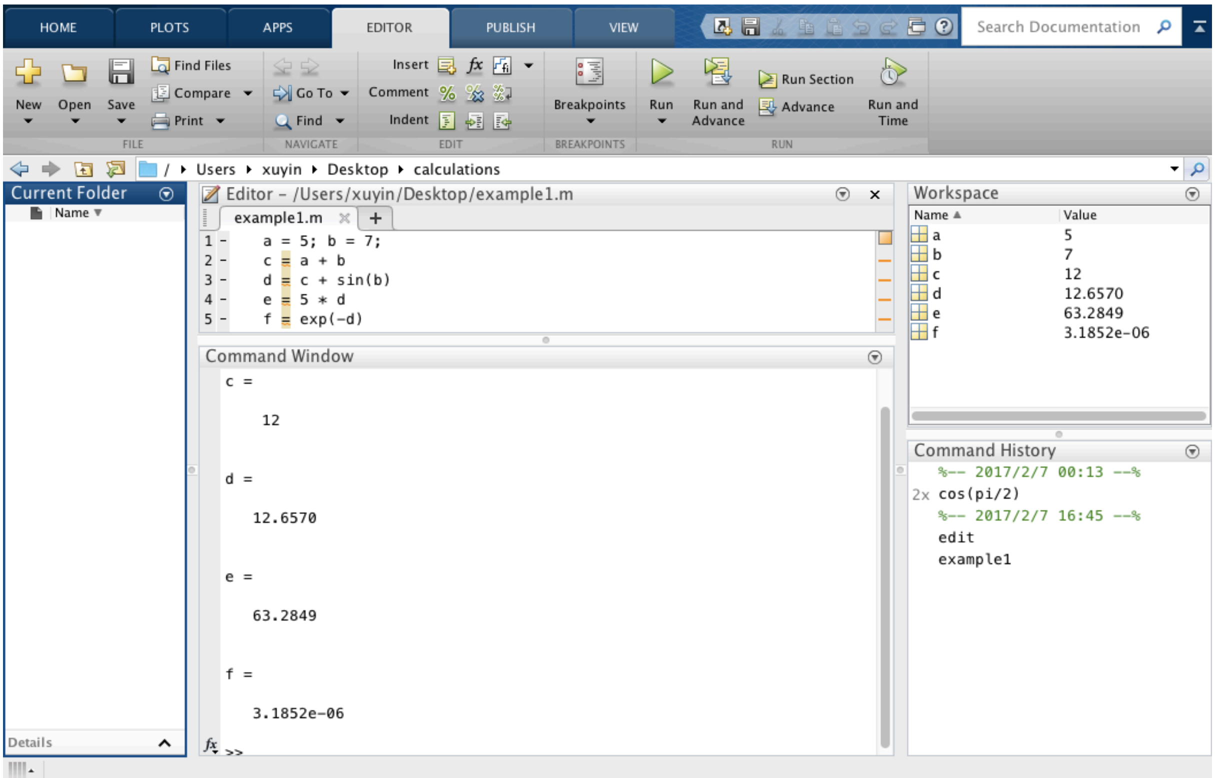
Command	Function
clear	Clear items from workspace
clearvars	Clear selected variables
disp	Display variable value
openvar	Open variable to workspace
who	List all variables in workspace
whos	List all variables in workspace, with sizes and types
load	Load variables to workspace
save	Save variables to file
matfile	Change variables into MAT-files,

Source: Moore (2010)

2.2.6 Editor Window

Sometime it maybe inconveniences to use the command window when writing lots of commands because every time we use the enter key, Matlab will run the code immediately. In these cases we need to use the M file editor. The command file and function file in Matlab is file with .m extension, so called M file. There are 3 ways to open the editor window. First is to use the toolbar in the command window. Second way is to choose New item in the File menu, then choose M-file. The third way is to use edit command in command window, see figure 2.5.

Figure 2.5 An example using editor



The editor is in the middle of the desktop. When use the editor, there will be a editor menu in the front of the window. In the editor widow, the main part is to write commands. The left side of the editor window is the number of command line which is automatically added by system. After finish the file, there are two ways to run the command. Click the Run button at the Editor menu or type the filename in the command window.

2.3 Program Structure of Matlab

In order to make a Matlab program, use the control structure is essential. The program structure of Matlab usually divided into 3 parts, sequential structure, selective structure and loop structure. Apply in these structures can make the commands more clear and increase the efficiency.

2.3.1 Sequential Structure

Sequential structure is the simplest structure. After finishing the command, system will execute it in order. These commands are easy to make with limited function. Figure 2.1 is a example of sequential structure.

Program 2.1 Example of sequential structure

```
>> a=[1 2 3];
b=[4 5 6];
c=a+b;

a =

     1     2     3
b =

     4     5     6
c =

     5     7     9
```

Source: Own calculation

2.3.2 Loop Structure

Loop structure is designed to solve the regular repetition calculation. Some part of the command will be run delicately. In every cycle the system needs to decide whether to run the loop according to the end condition. In Matlab, it provides for loop and while loop.

For loop is a control flow structure for chosen iteration. Fop loop executed repeatedly for specific times to reach the end keyword. Program 2.2 is a example of calculation of the sum of 1 to 10 by using for loop.

Program 2.2 Example of the for loop

```
sum=0;
for n=1:10;
    s=s+n;
end;
>> sum = 55
```

Source: Own calculation

While loop is running by given condition with unknown the number of iterations of cycle. The command is true when condition is fulfilled. Otherwise, the structure is false. If true, the cycle will continue to run the loop. The loop only ends when condition is not accomplished. Program 2.3 is an example of while loop for calculating sum of 1 to 10.

Program 2.3 Example of the while loop

```
sum=0;
n=1;
while n<=10;
   sum=sum+n;
   n=n+1;
end;
>> sum=55
```

Source: Own calculation

2.3.3 Selective Structure

Except for sequential structure and loop structure, Matlab also accept selective structure, which makes the process of programming more flexible and easy to use. There're two kinds of selective structure, which is if-else-end structure and switch-case-end structure.

The structure of the if-else-end structure is expressed as: IF (logical-expression) *statements a ELSE statements b* END. Statements a and b are consequence of the structure. If the logical expression is true, statement a will be run, else run statement b. Sometime we don't need to use else structure. Program 2.4 is a example of if-else-then structure.

Program 2.4 Example of if-else-then structure

```
a=100
b=20
if a<b
   'accept'
else
   'reject'
end
>> ans =
reject
```

Switch-case-end statement makes comparison of the logical expression and makes decision according to the result. Switch expresses some numeric variable or character variable. Checking the switch expression to match the case to execute. If the result is different from any case, execute otherwise or run out of the statement. An example of this structure is as program 2.5.

Program 2.5 Example of Switch-case-end statement

```
Month 1:10
switch month
   case { 1 2 3 };
      disp('Spring')
   case { 4 5 6 };
      disp('Summer')
   case { 7 8 9 };
      disp('Fall')
   otherwise
      disp('Winter')
end
>> month =
   10
Winter
```

2.4 Program Error and Debugger

In the process of making program, it is common to see errors in calculation. Usually the error includes 2 types, Syntax errors and execute errors. Executed error are anomalies case in program, for example, endless loop. Syntax error occurs when typing, for example, wrong spell of the function, missed brackets. These errors will stop the execution of programs. Matlab will show the error in the command window and return to the line with error. Program 2.7 is an example of error in Matlab.

Program 2.6 Example of error in Matlab

```
>> a=[1 2 3; 4 5 6];
>> b=[6 7 8; 9 1 0];
d=a*c
Undefined function or variable 'c'.
>> a*b
Error using  *
Inner matrix dimensions must agree.
```

Source: Own calculation

To debug a program, we can do the direct debug or using M-File editor. Direct debug is suitable for the simple situation. Firstly, if Matlab shows the line with error, users can check it directly about the brackets, function and some other problem. Secondly, user can run the program step by step to check.

Matlab provides some debugging commands that allow users to set, clear, and list breakpoints during the debugging process. Running M files by lines, examining variables in different workspaces, tracking and controlling program execution. Users can run debugging command in Editor. To debug a program, a function with a breakpoint is needed. When Matlab enters debug mode, the prompt is “K>>”.

Table 2.6 Debugger commands in Matlab

Command	Function
dbclear	Clear the breakpoints
dbcont	Resume the execution
dbquit	Quit the debug mode
dbstack	Display function call stack
dbstatus	List all breakpoints
dbstep	Run one or more lines from current breakpoint
dbstop	Set breakpoints
dbtype	List M-file with line numbers

Source: Bercher and Weeks (2007)

2.5 Graphing in Matlab

In Matlab, the command to connect the point into line is plot. Using one plot command can make multiple graphs in the same coordinate system. X1 and y1 are the numbers of the first line, parameter is the parameter option for the first line. X2 and y2 are the numbers of the first line, parameter is the parameter option for the second line. The parameter options determine the color, type and mark for data point. The commands are listed in the table below.

Table 2.7 Color options

Color	Command	Color	Command
red	r	pink	m
green	g	white	w
blue	b	black	k
yellow	y	light blue	c

Table 2.8 Type options

Type	Command	Example
actual line	-	
dotted line	--	
colon line	;	
point line	.	

Table 2.9 Mark for point of data

Command	Example	Command	Example
.	……	^	△△△△
+	+++++	<	◁◁◁◁◁
o	○○○○	>	▷▷▷▷
*	*****	p	☆☆☆☆
s		d	◇◇◇◇

Source: Bercher and Weeks (2007)

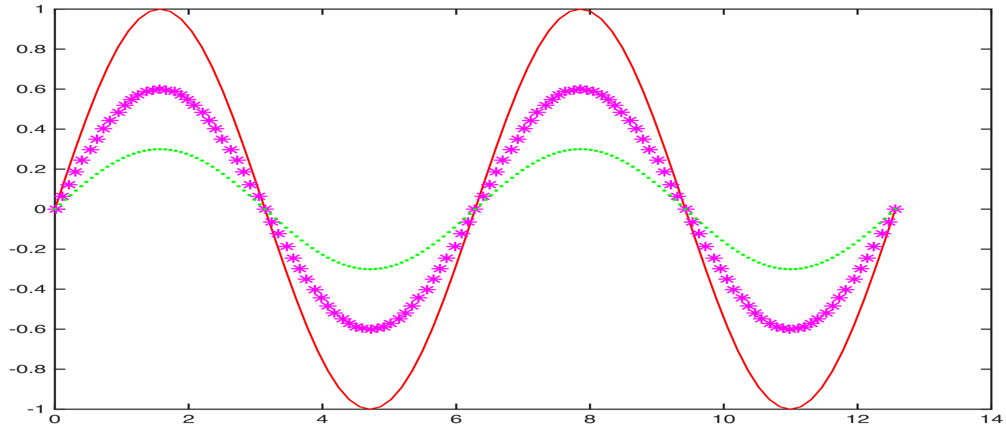
Users can use one or more settings for options for color, type and mark for data point. The symbol should be in one string. After plotting, some command can be applied to modify the plot. For example, user ‘grid on’ to add the grid line, use ‘title’ to add the title name and use ‘xlabel’, ‘ylabel’ to add label for x and y axis. Command ‘hold on’ can freeze the plot. When use ‘plot’ command again, the line will be added to the freeze plot. Figure 2.6 is an example for plotting in Matlab, using the command in program 2.8.

Program 2.7 Plot in Matlab

```
>> x1=0:pi/20:4*pi
x2=0:pi/30:4*pi
x3=0:pi/40:4*pi
y1=sin(x1)
y2=0.6*sin(x2)
y3=0.3*sin(x3)
plot(x1,y1,'-r',x2,y2,'*m',x3,y3,'.g')
```

Source: Own calculation

Figure 2.6 Example of plot in Matlab



Source: Own calculation

3. Description of Portfolio Optimization Models

Portfolio optimization was proposed my Markowitz（1952) and has developed for many years. Markowitz introduced a Mean-Variance model which mainly depends on mean and variance of return distribution. One kinds of work believed that the portfolio return follows normal distribution. After consideration of clustering of volatility and autocorrelation, returns are with fat-tails. To explain these facts, Mandelbrot (1963) and Fama (1965) put forward stable Paretian distribution. Then it was incorporated as a building block in GARCH-type processes. Another kinds of work focused on financial risks. For example, value at risk model was applied to estimate risks for potential high losses. Another type of model is called Conditional Value-at-Risk, also known as CVaR, is raised by Rockafellar and Uryasev (2002). The model is defined as the expected loss under the condition that it exceeds VaR.

In this chapter, we introduce the methodology of portfolio optimization, CVaR model and backtesting. The theory this chapter is mainly from Kresta(2015), Letmarka and Ringstrom (2006) and Hurlin and Tokpavi (2006).

Portfolio optimization theory is the basic theory in finance. The theory was created by Harry Markowitz in 1952. The work was followed by many economists, especially Merton, Samuelson and Fama. Economists then realized that this method could provide deep consideration into investment decision-making.

The problem is mostly formulated as automatically selected with random parameters within objective function. Objective functions usually have two parts. The first part is measure of expect utility function. From a theoretical point of view, the function is almost developed. It is based on axioms of substitution, transitivity ability and certainty equivalent. The models such as portfolio model and consumption model are based on this measure. We assume the random variables are normally distributed or the utility function is quadratic or can be considered approximately as Taylor series expansion. These kinds of problems we can formulate as mean-variance model. It is sufficient to describe the probability distribution just by means of two parameters, mean value and variance. The second type can be simply called “safety first”. These are managerial criterions that are typical by the aim that the selection of a portfolio eliminating extreme lose events. For example, value at risk, risk adjusted return on capital.

3.1 Risk Measures

Risk measurement, also known as risk evaluation is a way to identification and quantitative of risk based on the identification of risk. It is a way to analyze and predict risk to evaluate the possibility that the risk accident occurs and the loss that the accident may cause. Risks derived from the uncertainty of future events. It shows the possibility of varieties of results. We want to find compensation for the investment. This measurement enables for calculation of relatively accurate probability of loss that can eliminate the uncertainty of the loss. The forecast of the loss can make investors understand the consequences of the risk of the loss and focus on the consequence the risk brings. We define the risk measure $\square(X)$ as a mapping function from random variable to real numbers. From financial view, we want to find the coherent risk measures, which was raised by Artzner at al. (1999). The theory proposed that a well-defined risk measure should follow four axioms, which are monotonicity, positive homogeneity, translational invariance and sub-additive. The risk measures that meet these axiom is called coherent risk measure. The axioms are defined as:

1. translational invariance: we define A as a portfolio then $\rho(X+a)=\rho(X)-a$, which we add cash a to X portfolio, the amount of value decrease risk by same amount,
2. monotonicity: for $X_1 \geq X_2, \rho(X_1) \leq \rho(X_2)$. The portfolio with better return is always with lower risk.
3. sub-additivity is $\rho(X_1+X_2) \leq \rho(X_1)+\rho(X_2)$. It implies that risk of two portfolios together can't exceed the situation when the risk is separately, which means the diversification is necessary,
4. positive homogeneity is $\rho(a \cdot X) \leq a \cdot \rho(X)$. It is the situation that we increase a times of the portfolio, we will get a time increase in risk.

3.1.1 Standard Deviation

Standard deviation is a way to measure the statistical dispersion of the data from the mean. It reflects the degree of discrepancy and distribution of data. The standard deviation is positive. Mathematically it is the square root of variance. Variance measures the risk of a portfolio. It is the expectation of the squared deviation from the mean. The variance of a portfolio equal to the weighted average covariance of the returns on its individual securities:

$$\mathrm{Var}\left(r_p\right)=\sigma_p^2=\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \mathrm{Cov}\left(r_i, r_j\right), \tag{3.1}$$

where

$$\text{Cov}\left(r_i, r_j\right) = \rho_{ij}\sigma_i\sigma_j = \sigma_{ij} \quad , \tag{3.2}$$

when ρ_{ij} is correlation coefficient of the portfolio. So variance can be calculated as

$$\text{Var}\left(r_p\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \rho_{ij}\sigma_i\sigma_j \ . \tag{3.3}$$

For portfolio optimization, we calculate the variance based on Markowitz framework. We assume the portfolio consists of *N* numbers of assets. The expect return of asset is defined as $E(R) = \{E(R_1),...,E(R_n)\}^T$. The covariance matrix is as $Q = \{\sigma_{i,j}, i = 1,...,N, j = 1,...,N\}$. Then we assume the portfolio composition $x = \{x_1,...x_n\}^T$, we can further define the expect return and portfolio variance as follows:

$$E(R_p) = \sum_{i=1}^{N} x_i \cdot E(R_i) = x^T \cdot E(R) , \tag{3.4}$$

$$\sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot \sigma_{i,j} \cdot x_j = x^T \cdot Q \cdot x , \tag{3.5}$$

$$\sigma_p = \sqrt{\sigma_p^2} \quad . \tag{3.6}$$

There are advantages of standard deviation. The standard deviation is a best measure of variation. It shows the correlation of the data around the mean value and it use every item of dataset. It measures the level how the data is distributed and not affect by extreme values. Also standard deviation allows combination of two groups. The disadvantage of standard deviation includes asymmetric payoffs. Standard deviation measures two side risks. However in finance, the risk is connected to the losses.

3.1.2 Value at Risk

Generally the risk is resulting from the instability of the financial market, which will cause high volatility of returns thus we need to eliminate financial risks. The risks mainly from the market risk, which includes credit risk, stock risk, foreign exchange risk, commodity risk, interest rate risk and option risk. VaR was developed to state the risk level. Value at risk, also VaR, is a current international standard risk management tool proposed by Morgan (1980) for bank's business risk. VaR refers to the potential maximum loss of a portfolio under a given confidence level in the chosen period. VaR is a quantification and measurement of the financial risk based on the probability theory and mathematical statistics. It can derive a one-dimensional approximate value of the multidimensional risk, which can be applied to various markets. The Basel Committee on Banking Supervision, the Federal Reserve Bank, the US Securities and Exchange Commission, and the European Union accept VaR as a tool for risk measurement and risk disclosure. VAR can be defined as follows,

$$VaR_{X,\alpha} = -inf\{ x \in R : F_X(x) \geq \alpha \}, \tag{3.7}$$

where α is the probability level that specifying the possibility that the observed loss can be exceed estimated VaR,

$$Pr(X \leq -VaR_{X,\alpha}) = \alpha \;. \tag{3.8}$$

From the calculation of the VaR we can see that we need to determine 3 variables, which are confidence level, portfolio return and the period we hold them. The confidence level and the holding period is chosen by the investor. Mostly we use α level at 15% based on Solvency II, 5%, which is original applied by JP Morgan, 1% according to Basel II and 0.5% based on Solvency II.

3.1.3 Conditional Value at Risk

Conditional value at risk (CVaR) is an risk measurement method developed on the basis of VaR (Value at risk). As a risk measurement method, VaR have simple concept that is simple and easy to communicate and understand. It provides a unified, comprehensive risk measurement framework for complex portfolio of different financial instruments. In practice, we found many problems of the VaR model. In order to improve the shortcomings of VaR, Rockafeller and Uryasev proposed conditional value at risk measurement techniques in 2000. CVaR refers to the average loss of the portfolio under the condition that the loss of the portfolio is greater than a given VaR value. CVaR is sub-additive, positive homogeneous, monotonicity and transmit invariance. CVaR is a consistent risk measurement method and can be optimized by using a linear programming algorithm. CVaR is given more and more attention of institutional investors. Generally, CVaR is calculated as:

$$CVaR_{X,\alpha} = -E[x \mid x < -VaR_{X,\alpha}] \,, \tag{3.9}$$

where X is the random-variable profit and x are the realizations of X. We assume that we have the future possible profits X with equal probability, CVaR can be defined as

$$CVaR_{X,\alpha} = -\frac{1}{\alpha} [\frac{1}{n} \sum_{\alpha=1}^{[\alpha n]-1} X_\alpha + (\alpha - \frac{[\alpha n]-1}{n} X_{[\alpha n]}] \,, \tag{3.10}$$

where *[x]* represents for smallest integer that is larger than *x*. Variable *n* is defined as the quantity of data utilized for CVaR.

3.2 Naive Strategy

Naive strategy is a strategy to split the wealth of the investor uniformly between the available investment possibilities. This strategy can be traced back to 4th century, where Rabbi Issac bar Aha gives the advice as " *One should always divide his wealth into three parts: a third in land, a third in merchandise, and a third ready to hand"*.[1] When individual investors make investment decision, they seldom choose difficult strategies. Thus 1/n strategy is wildly used. In this situation, the weight of the asset for the portfolio is equal, which is

$$w_t^{ew} = 1/N \quad , \tag{3.11}$$

where N is the number of assets. This strategy doesn't consider the optimization and estimation and ignore the influence of data.

3.3 Description of Mean-Variance Model

There're 6 assumptions of mean-variance model, which includes:

1. The mean-variance model is a static model. There's maximum one period of the expect utility and utility curve,
2. Investors only invest in risky assets,
3. Investors are risk aversion,
4. The market is information efficient,
5. There are negation of transaction costs and taxes,
6. Investors are able to combine any composition. There's no limit of assets. The application of the mean-variance model includes chosen of the feasible and efficient set and chosen of the optimal portfolio.

In general, it is assumed that the investor measures the return level by calculating the expected value of the distribution. So we use the probability distribution of the expected return of the portfolio. Also, we assume that the risk can be measured by the variance around the expected return of the probability distribution. The most acceptable measure of this variability is variance and standard deviation.

[1] Babylonian Talmud: Tractate Baba Mezi'a, folio 42a.

We assume a set of risky assets and weights that describes portfolio investment is split, the general formulas of expected return for *n* assets is

$$E(r_P)=\sum_{i=1}^{n} w_i E\left(r_i\right), \tag{3.12}$$

where $\sum_{i=1}^{n} w_i$ equals 1. Variable *n* is the number of securities. Variable w_i equals the proportion of the funds invested in security *i*. Variable r_i, r_P equals the return on i^{th} security and portfolio *p* and $E(r_P)$ is the expectation of the variable in the parentheses. The computation of portfolio return is to find the weighted average return of the securities included in the portfolio. We further assume that the investor's risk aversion is given. The utility function of the investor is as followed:

$$U = E(r_p) - k \cdot \sigma_p^2 . \tag{3.13}$$

Under this situation, the investor gets maximize expected return from the portfolio and minimum risk. The risk-return that the investor could undertake with the same utility is decided by parameter *k*. Different risk attitudes investors have different the value of parameter k. For risk averse investor, parameter k is positive while risk seekers are with negative value. For risk neutral investors, the parameter is zero. Efficient portfolios contain numbers of asset combinations, which is defined as formula 3.14

$$v=\left\{\begin{array}{l} arg\, min\, k \cdot x^T \cdot Q \cdot x-(1-k)\cdot r^T \cdot x \\ \sum_{i=1}^{n} x_i = 1 \\ x_i \geq 0, \quad i=1,\ldots,n \end{array}\right\}. \tag{3.14}$$

If we know the different *k* level, we can obtain different optimal portfolio composition. If *k*=0, the results is equal as the maximum return of the portfolio that we don't consider the risk level. As $k \in [0,1]$, we can get different portfolio optimization strategies.

3.4 Description of Mean- CVaR Model

For investors, the aim of investment is to seek the maximize returns with minimum risk. We assume that the returns of portfolio follow heavy tail distribution, such as t distribution. We can get the optimization function of the mean-CVaR model as

$$v = \begin{Bmatrix} \arg\min k \cdot CVaR_\alpha \cdot (\mathbf{R} \cdot x) - (1-k) \cdot E(\mathbf{R} \cdot x) \\ \sum_{i=1}^{n} x_i = 1 \\ x_i \geq 0, \quad i = 1,...,n \end{Bmatrix}, \tag{3.15}$$

where $\boldsymbol{R}$ is matrix of returns, with the columns represent assets and rows represent scenarios. The risk estimation is replaced by CVaR. If k=0, the result is the same as mean-variance model, as we don't consider the risk. Generally we use $k \in [0,1]$ for calculation of optimal portfolio.

3.5 Backtesting of Portfolio Optimization

The goal of portfolio optimization is to maximize return and minimize risk by changing the distribution of capital investment. However, these parameters are unknown when formulate and solve the problem. These might lead to large error. Therefore, backtesting can be applied in the model. Within the backtesting procedure, historical data are utilized. We would calculate the portfolio allocation based on the information of each observation. The weights of portfolio at time t are based on returns of the assets in the period ($t - m$, $t - 1$). Variable m is the size of past data. Based on these we are able to get the ex post portfolio return $r_{P,t}$.

$$r_{P,t} = \sum_{i=1}^{n} r_{i,t} \bullet W_{i,t} \tag{3.21}$$

Where $r_{i,t}$ is the ex post returns, $W_{i,t}$ is the weights of assets, which can be calculate as:

$$W_{t+1} = W_t \cdot (1 + r_{P,t}) \tag{3.22}$$

If we assume the wealth path $\{W_t\}_{t=0}^{t}$ at the time $\tau \in (0,T)$. We can calculate the drawdown, which is the decline from history maximum peak. The decline size at time τ depends on the highest peak,

$$DD_\tau = 1 - \frac{w_\tau}{\max_{t \in (0,\tau)} w_t} \tag{3.23}$$

We can extend the ratio to measure the maximum drawdown over a period. Maximum drawdown is the largest decline in the chosen period.

$$MDD_{0,T} = \max_{\tau \in (o,T)} \left[DD_\tau \right] \tag{3.24}$$

Also, Sharpe ratio could be applied to measure the risk adjusted return. It is the ratio between excess expected return. If the Sharp ratio is higher, the return relative to risk is higher. Sharpe ration is calculated as:

$$Sharpe\ ratio = \frac{E\left(R_P\right) - R_f}{\sigma_P} \tag{3.25}$$

Where $E\left(R_P\right)$ is the expected return, R_f is the risk free rate and σ_P is the standard deviation.

4. Portfolio Optimization Backtesting in Matlab

In this chapter we use the methodology described in chapter 2 and 3 to calculate the portfolio optimization strategy. We choose 43 stocks from Hong Kong stock exchange from 2006 to 2016 apply 3 different methods, which are naive strategy, Markowitz model and mean-CVaR model. We use the initial wealth 1 HKD and compare the result of our investing. We calculate the Markowitz model and mean-CVaR model of different k value and α value. Also we divide the data into in-sample and out-of-sample and calculate both simple investing based on in-sample data or rolling window principle to see the difference of the model.

The chapter is divided into 6 sections. In the first section we start with data acquisition and calculation of returns. We choose 43 stocks, which are components of Hang Seng Index and list the name of the stocks. Then we divide the stocks to 4 parts based on the stock price in the start date and evaluation the change and situation of stocks. In the second section we apply the naive strategy, invest equal weight to stocks. In section 3, we generate the feasible set and find the efficient frontier. Then in the fourth section we apply the Markowitz model with different values of k and the rolling window strategy. In the fifth section we apply CVaR model and also do the comparison based on different α levels. Then we do the comparison of the models.

4.1 Data Description

The goal of the thesis is to analyze the optimal portfolio of CVaR model and make comparison to Markowitz Mean-Variance Model. For the analysis the first step we need to do is data acquisition. For calculation, we choose 43 stocks from Hang Seng Index that have complete data from 2006 to 2016 and are listed in Hong Kong Exchanges (HKEx). Table 4.1 lists the names of the chosen stocks. We use the financial data and time series from finance.yahoo.com website.

Table 4.1 Lists of Companies

Names	Abbreviations	Names	Abbreviations
Bank of East Asia	BEA	Henderson Land	HLD
Bank of Communications	BC	Hengan	HGI
Belle International	BI	HK&CHINA Gas	HKAG
Bank of China	BOC	HK Exchange	HKEX
Cathay Pacific Airline	CPA	HK& Shanghai Banking	HSBC
China Construction Bank	CCB	Kunlun Energy	KE
China Life Insurance	CL	Lenovo	LEN
China Merchants Port	CMH	Li& Fung	LF
China Mobile Limited	CM	Mengniu Dairy	MN
China Overseas Land& Investments	CO	Mass Transit Railway	MTR
		Hang Seng Bank	HSB
China Resource Beer	CRB	New World	NW
China Resources Land	CRL	Petro China	PC
CHINA RES POWER	CRP	Ping An	PA
China Resources Power	CAP	Power Assets	PAS
China Unicom	CU	Sun Hung Kai Properties	SHK
CITIC	CIT	Sino Land	SL
Cheung Kong Hutchison	CKH	Sinopec	SC
China Light & Power	CLP	Swire Pacific	SP
China National Offshore Oil	CNO	Tencent	TEN
Galaxy Entertainment	GE	Tingyi	TI
Hang Lung Properties	HLP	Wharf	WHA

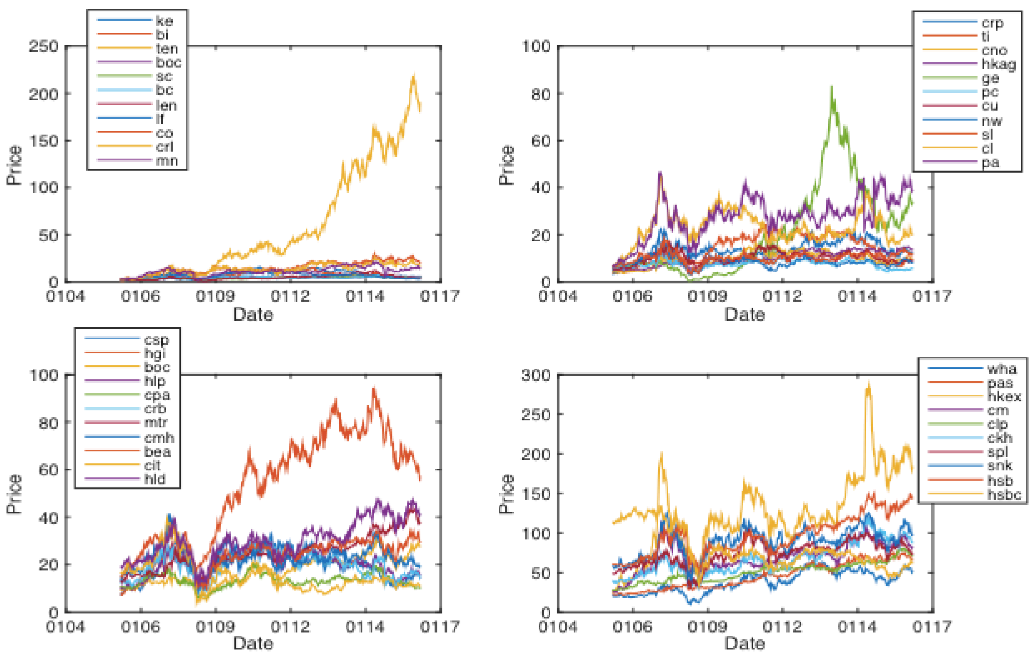
The stocks we choose for our portfolio is from Hang Seng Index. Hang Seng Index is an important indicator of price of Hong Kong stock market. The index chooses is calculated from the market capitalization of constituent stocks. The index includes Hang Seng Financial Sub-Index, Hang Seng Utilities Sub-Index, Hang Seng Properties Index and Hang Seng Industrial and Hang Seng Commercial Sub-Index. It represents more than 70% of market share in Hong Kong stock Exchanges. Hang Seng Index was first public on November 24th, 1969 with the based period from July 31st and the base index 100. It is the most influential price index that reflects the market trend in Hong Kong market. For investors, Hang Seng Index mainly records the daily changes in the market and reflects the overall performance of market. The index is also a portfolio that simulated investment based on the market share.

Table 4.2 Mean and standard deviation for return of chosen stocks (weekly)

	Whole period		In-sample period		Out-of-sample period	
	Mean	Std.	Mean	Std.	Mean	Std.
BC	0.251%	4.824%	0.320%	5.598%	0.168%	3.697%
BEA	0.214%	4.360%	0.302%	5.245%	0.110%	2.980%
BI	0.391%	6.804%	0.925%	7.933%	-0.243%	5.078%
BOC	0.266%	3.907%	0.251%	4.618%	0.283%	2.840%
CAP	0.301%	4.582%	0.394%	5.465%	0.190%	3.230%
CCB	0.241%	5.990%	0.383%	7.362%	0.071%	3.751%
CIT	0.164%	6.410%	0.263%	7.893%	0.046%	3.985%
CKH	0.222%	4.002%	0.212%	4.722%	0.234%	2.925%
CL	0.335%	5.298%	0.484%	5.771%	0.157%	4.669%
CLP	0.189%	2.225%	0.249%	2.600%	0.119%	1.670%
CM	0.270%	3.847%	0.386%	4.588%	0.132%	2.710%
CMH	0.296%	5.486%	0.522%	6.478%	0.027%	3.985%
CNO	0.516%	6.229%	0.677%	7.373%	0.326%	4.499%
CO	0.073%	3.992%	0.136%	4.440%	-0.002%	3.384%
CPA	0.225%	6.124%	0.386%	6.090%	0.033%	6.159%
CRB	0.521%	6.831%	0.712%	8.065%	0.294%	4.972%
CRL	0.378%	5.584%	0.611%	6.300%	0.102%	4.574%
CRP	0.284%	5.783%	0.650%	6.741%	-0.151%	4.343%
CU	0.210%	5.149%	0.478%	5.805%	-0.107%	4.219%
GE	0.590%	7.212%	0.692%	8.521%	0.469%	5.246%
HGI	0.443%	4.190%	0.813%	4.872%	0.002%	3.139%
HKAG	0.228%	3.191%	0.286%	3.664%	0.159%	2.516%
HKEX	0.490%	5.417%	0.675%	6.463%	0.270%	3.806%
HLD	0.244%	4.669%	0.204%	5.416%	0.292%	3.584%
HLP	0.210%	5.064%	0.383%	6.109%	0.005%	3.422%
HSB	0.205%	3.248%	0.141%	3.990%	0.281%	2.040%
HSBC	-0.038%	3.640%	-0.153%	4.185%	0.099%	2.854%
KE	0.370%	5.614%	0.782%	6.583%	-0.119%	4.127%
LEN	0.289%	6.359%	0.420%	7.470%	0.134%	4.707%
LF	0.161%	5.427%	0.541%	6.062%	-0.290%	4.518%
MN	0.537%	7.348%	0.588%	7.024%	0.477%	7.715%
MTR	0.230%	2.959%	0.230%	3.548%	0.230%	2.050%
NW	0.196%	5.406%	0.118%	6.400%	0.288%	3.907%
PA	0.548%	7.103%	0.626%	6.752%	0.456%	7.498%
PAS	0.217%	2.480%	0.258%	2.590%	0.167%	2.341%
PC	0.134%	4.829%	0.301%	5.540%	-0.065%	3.806%
SC	0.280%	4.875%	0.427%	5.668%	0.106%	3.713%
SHK	0.188%	4.381%	0.252%	5.255%	0.111%	3.030%
SL	0.257%	5.631%	0.341%	6.919%	0.157%	3.533%
SP	0.142%	3.689%	0.243%	4.372%	0.021%	2.655%
TEN	0.966%	5.441%	1.121%	6.507%	0.781%	3.797%
TI	0.267%	4.940%	0.697%	5.514%	-0.244%	4.098%
WHA	0.274%	4.829%	0.278%	5.708%	0.269%	3.510%

To do the calculation, we choose 11 years of weekly-adjusted data from 16th January 2006 to 26th December 2016. So we have data for 572 weeks. In table 4.2 the greener color represents better data. We can see that TEN is with highest mean return for whole period. The lowest mean return for the whole period is HSBC. The lowest standard deviation of return for whole period is PAS. For in-sample period, TEN had highest mean return and GE had highest standard deviation. Moreover, we can run program A1 in Annex and get figure 4.1. It shows the evolution of the stocks. We assume the data of the first 6 years, which is from 2006 to 2011 as the in-sample data and from 2011 to 2016 the data is the out-of-sample data. The initial wealth is 1 HKD.

Figure 4.1 Evolution of chosen stocks



In order to show the series more clearly, we divide the stocks to 4 parts based on the price at 16th January, 2006. We can see from the upper left that nearly all of the stocks increased during the past 11 years. There's a huge increase of Tencent. The price is nearly 104 times more than it was in 2006, with the lowest price 1.8 HKD and the highest price 218.2 HKD. Tencent is earliest instant messenger application developer. It created the application QQ. With the rapid development of Internet technology, Tencent upgrade their service and create more services like various kinds of online games, online shopping. The instant messenger application Wechat now has more than 350 million users. All these factors caused the rise in the stock price of Tencent.

From the upper right figure, we can see that the chosen 11 stocks are all with an increase tendency. The highest growth among these stocks is from the Galaxy Entertainment Group. Galaxy Entertainment is one of Asia's lending developers and operators of integrated entertainment. Galaxy Casino in owned by the group. Galaxy Casino is a gaming concessionaire that received a gaming concession of Macau government. Only 3 companies had the gaming concession of the government. For the license, the company spent 16.7 billion HKD and these amounts of money were calculated as intangible asset amortized from the profit. Because the casino opened in 2002, there're initial expense and expenditure for new projects, which cause the bad financial situation. The price decrease from 5 to 0.55 HKD in 2008. Then the company revalued their intangible asset and changes this situation. The highest price occurred in 2014, which was 82.65. However, to restrain the development of the industry, the gaming tables are limited to 5500 in total industry, which lead to the decrease in further period.

In the lower left figure, CITIC and China Resources Power had decreased. The stock with extreme in these years is Hengan International. It is the earliest enterprise in sanitary napkins market in China. In these years it extends their business to baby & adult diapers, tissue paper & wet wipes etc. Hengan has been the market leader with the 1st market share in these years.

For the lower right figure, except for HK& Shanghai Banking had a little decreased in 2016 than the price it did in 2006, all stocks increased their price on the bases of 2006. From 2009 to 2015, the HK Exchange expanded for more than 10 times.

Also we can get the 10-year government bond rate is 2.84%. With the initial price 1 HKD, the weekly risk free rate government bond could be calculate as:

$$Weekly\ risk\ free\ rate = 1 \cdot (1+2.84\%)^{\frac{1}{52}} - 1 = 0.0539\%. \tag{4.1}$$

4.2 Portfolio Optimization of Naive Strategy

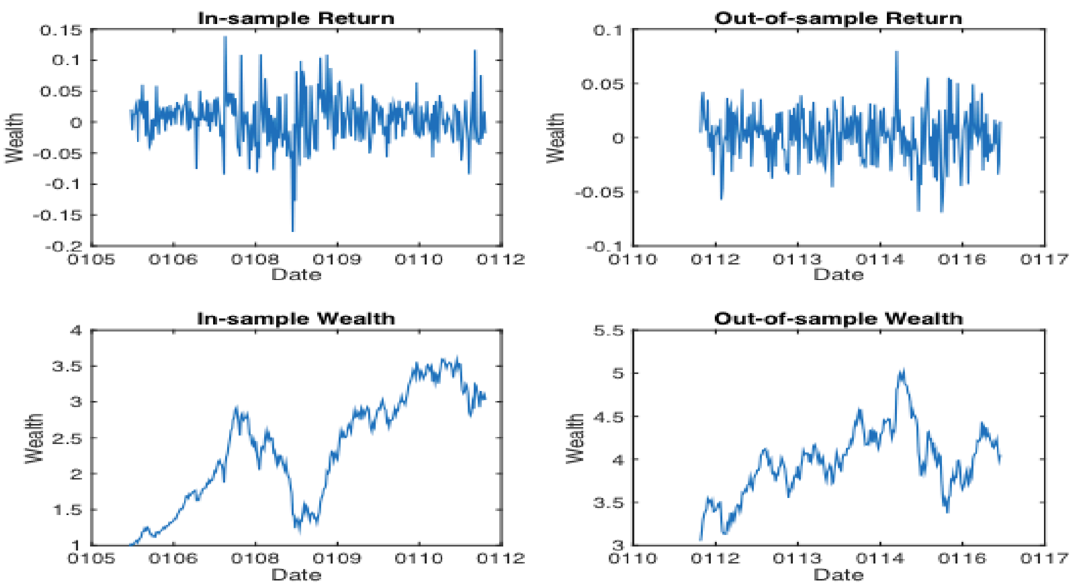
Naive strategy is a rough and instinctive common sense division of a portfolio. We assume the weight that investors apply to each stock is the same. In our case, the weight of each stock is $\frac{1}{43}$. Running the program A2, we get the result for naive strategy as Table 4.3.

Table 4.3 Results for Naive Strategy

Date	Return	Wealth	Date	Return	Wealth
2006/1/23	0.0196	1.0196	2012/1/2	0.0051	3.0580
2006/1/30	-0.0133	1.0061	2012/1/9	0.0345	3.1636
2006/2/6	0.0120	1.0182	…	…	…
…	…	…	2016/4/25	-0.0218	3.8992
2007/11/5	-0.0384	2.8070	2016/5/2	-0.0450	3.7238
2007/11/12	-0.0446	2.6818	…	…	…
…	…	…	2016/10/3	0.0220	4.3742
2007/12/31	0.0172	2.8267	…	…	…
2008/1/7	-0.0223	2.7638	2016/12/5	0.0124	4.2298
…	…	…	2016/12/12	-0.0341	4.0857
2011/12/19	0.0176	3.0952	2016/12/19	-0.0235	3.9898
2011/12/26	-0.0170	3.0425	2016/12/26	0.0140	4.0459

From table 4.3 we have the result for naive strategy. The initial wealth is 1 HKD. In the left side we have the in-sample result and the right side is the out-of-sample result. In naive strategy, we use the constant weight for investing regardless of the change of the situation of the stock. For in-sample data, the final wealth is 3.04, which means we earn 2.04 for our portfolio. For out-of-sample result, the final wealth is 4.04. In this period, we earn 1 HKD. To represent the result more distinctly, we plot the result in figure 4.2.

Figure 4.2 Result of Naive Strategy



From in-sample result, most of weeks we have positive portfolio returns, with the maximum number 0.139. Only in 77 weeks among 311 we get negative returns. The minimum return is 0.177 negative. The negative results mainly occurred in 2008. The global financial crises made some influence on Hong Kong stock market. In 2008, most of the stocks we choose have a decrease in price, which lead to a decrease in our wealth. After 2008, the in-sample results was increased. For our-of-sample results, the portfolio returns fluctuate between 0.05 and -0.05. Nearly half of the returns is negative. In 2015 there were a devaluation of HKD and large holders decreased holding of stocks, which leads to a decrease in stock price.

Based on these results, we could also measure the performance of the chosen portfolio. In our case, Maximum drawdown and Sharp ratio is calculated as table 4.4.

Table 4.4 Performance of naive strategy

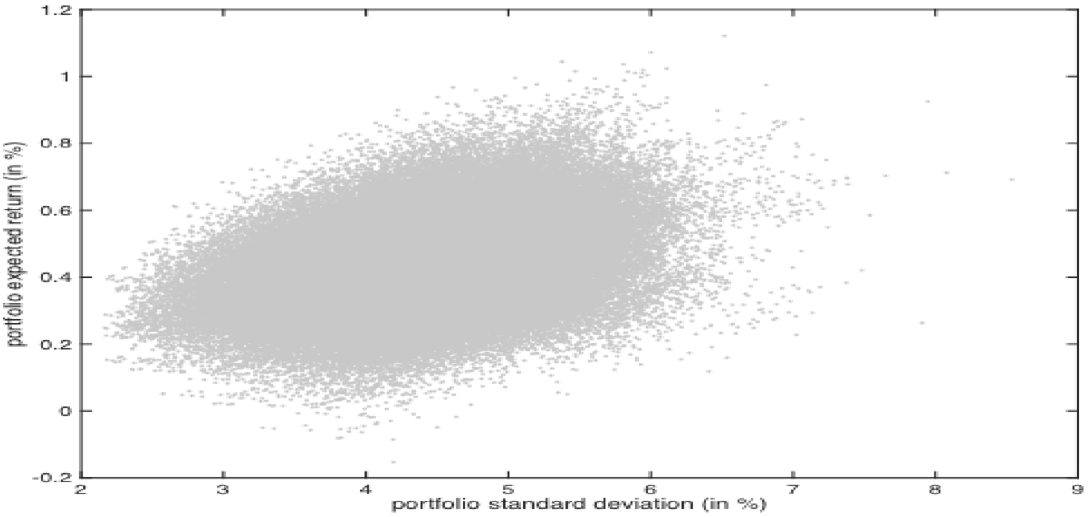
	In-sample	Out-of-sample
Final wealth	3.043	4.046
Return	204.25%	32.30%
Annual Return	20.45%	5.74%
Std. (annual)	27.54%	16.70%
Maximum drawdown	57.91%	32.66%
Sharpe ratio	9.93%	3.48%

From table 4.4 we can see that the results for both in-sample and out-of-sample data. The return rate of in-sample period is significantly higher than out-of-sample period. The final wealth of out-of-sample period is higher than in-sample period and the standard deviation is lower. The in-sample data is with higher maximum drawdown and Sharpe ratio.

4.3 Generation of Feasible Set

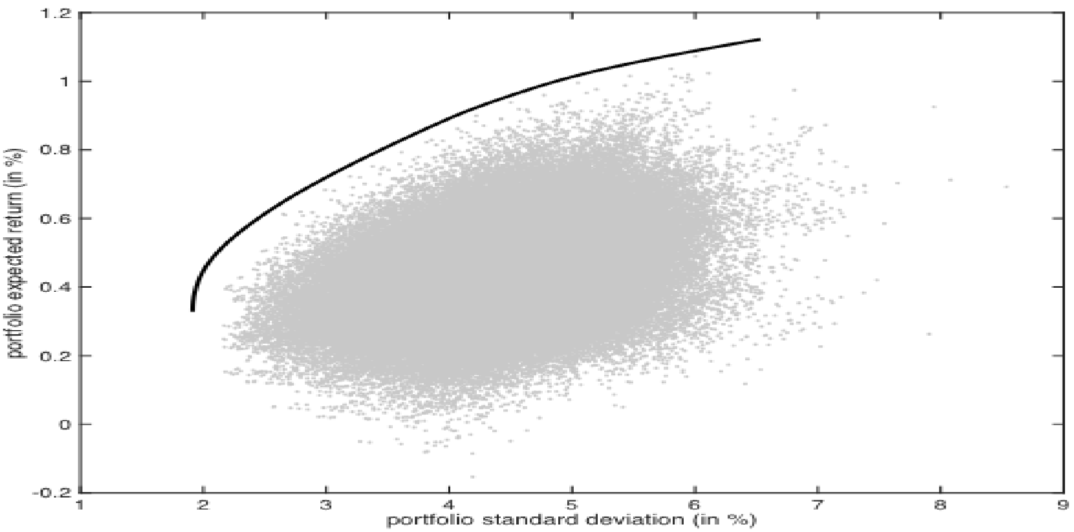
First of all, we assume that all of the possible portfolios can be invest. Only risky assets can be invested and no short selling is allowed. Based on in-sample data we can calculate the expect returns of portfolio. Then we can apply program A3 to calculate and plot the efficient set. We start the calculation with calculate the average returns and covariance of the portfolio. Then generate all possible vectors of weights and assume only expected returns and variance as portfolio parameters. The calculation is based on in-sample data. The result is as figure 4.3.

Figure 4.3 Feasible set of portfolios



We set the portfolio interval as 0.25 and we can get the result as figure 4.3. However, we are also interested in the best result, which is the optimal portfolio. The result named efficient set, which isbased on investor's risk aversion. In our case we can calculate the efficient set as figure 4.4

Figure 4.4 Efficient set of portfolios



From figure 4.4 we can see that the efficient sets standard deviation varies from 1.5% to 7%. The portfolio return is between -0.2% and 1.2%.

4.4 Portfolio Optimization of Markowitz Model

For Markowitz model, we assume that the investors are risk averse, which means that if there exist two portfolios with the same expected return, investors will prefer the less risky one. In this chapter, we calculate the backtesting of Markowitz model from 2 different approaches.

4.4.1 Backtesting of Markowitz Model with Different *k* Values

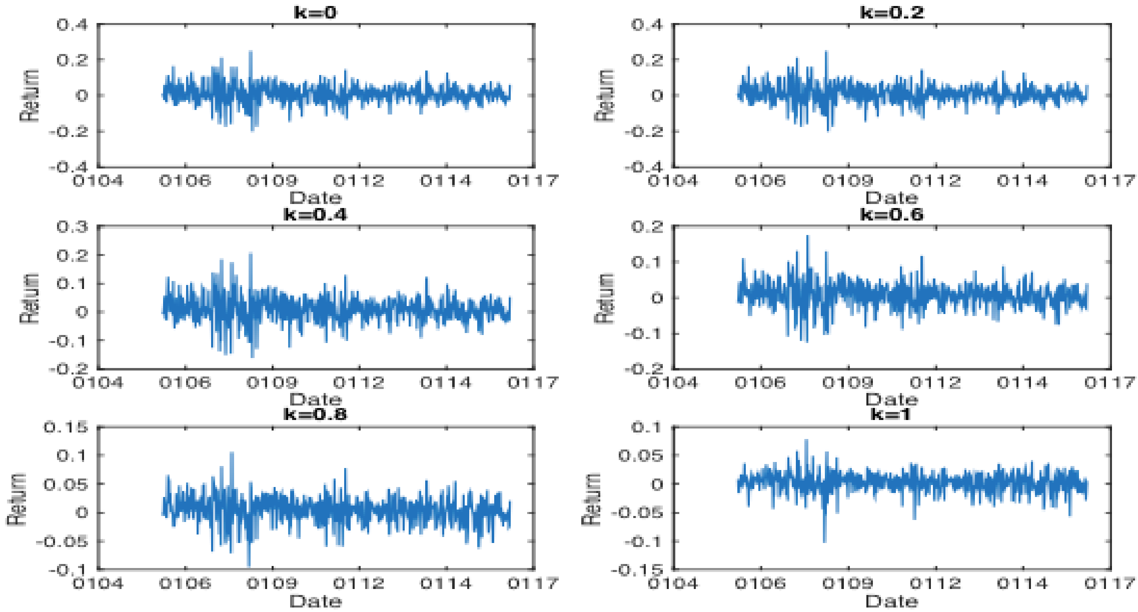
For calculation, we divided our data into two parts. From 2006 to 2011 are in-sample data and from 2012 to 2016, data is out-of-sample. We simply calculate the result for 2 parts on different values of *k*. In this way of backtesting, we obtain weights from the in-sample data and invest in this weight from 2012 to 2016. The chosen *k* level is 0, 0.2, 0.4, 0.6, 0.8 and 1. By running the program A4 in Annex A, we can get the result for this model.

Table 4.5 Calculation of Markowitz model with different *k* values

Dates	k=0.2		k=0.4		k=0.6		k=0.8		k=1	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2006/1/23	0.50%	1.01	-0.57%	0.99	-1.25%	0.99	-0.14%	1.00	0.65%	1.01
…	…	…	…	…	…	…	…	…	…	…
2008/6/10	-9.21%	6.33	-9.43%	5.67	-8.65%	4.54	-5.32%	2.97	-2.94%	1.83
2008/6/16	1.29%	6.41	1.67%	5.76	1.45%	4.60	0.86%	3.00	0.52%	1.84
…	…	…	…	…	…	…	…	…	…	…
2011/12/19	4.13%	16.90	3.58%	16.58	2.60%	13.68	1.47%	6.89	0.47%	2.65
2011/12/26	-1.82%	16.59	-1.83%	16.28	-1.70%	13.44	-0.73%	6.84	-0.30%	2.64
2012/1/2	-1.03%	16.42	-1.01%	16.11	-0.31%	13.40	0.58%	6.88	0.96%	2.67
2012/1/9	10.10%	18.08	7.39%	17.31	3.92%	13.93	1.21%	6.97	-1.57%	2.62
…	…	…	…	…	…	…	…	…	…	…
2015/3/16	6.74%	78.57	5.87%	58.82	3.87%	30.04	2.08%	11.24	0.56%	3.38
2015/3/23	-0.49%	78.19	0.49%	59.11	1.00%	30.34	1.34%	11.39	1.85%	3.44
…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-2.28%	99.53	-2.73%	65.31	-3.07%	26.85	-2.43%	9.58	-1.63%	3.13
2016/12/26	5.56%	105.07	5.05%	68.62	3.77%	27.86	1.96%	9.76	0.35%	3.14

From table 4.5 we can see the result of calculations of Markowitz model based on different *k* values. The result of *k* values is expressed as percentage form. When *k=0*, we receive the result that the goal is only to maximum return, regardless of the risk. In our case, after calculation we get the result of *k* equals to 0 is the same as *k* equals to 0.2. Also because when *k* equals to 0 we didn't consider the risk, the results of apply Markowitz model or CVaR model is the same. The highest return is when *k*=0 and *k*=0.2, where out-of-sample wealth at the end is nearly 9 times more than beginning of the period. The final wealth is 105.07. When *k*=1, the final wealth is 3.14. Figure 4.5 shows the return of the portfolio.

Figure 4.5 Return of Markowitz portfolio based on different levels of *k*



We can see from figure 4.5 when *k* is higher, the portfolio returns are increasing. When *k*=0, the highest portfolio return is 0.4. When *k*=1, this number decrease to 0.1. However, the possibility to get negative returns in weeks is also higher. When *k*=0, the minimum return is 0.4 negative. When *k*=1, the minimum return is -0.15. The returns of the portfolio are more stable. In the in-sample period, the data are mode variable. In the year 2008, the portfolio return fluctuated more frequently.

Figure 4.6 Wealth of Markowitz model

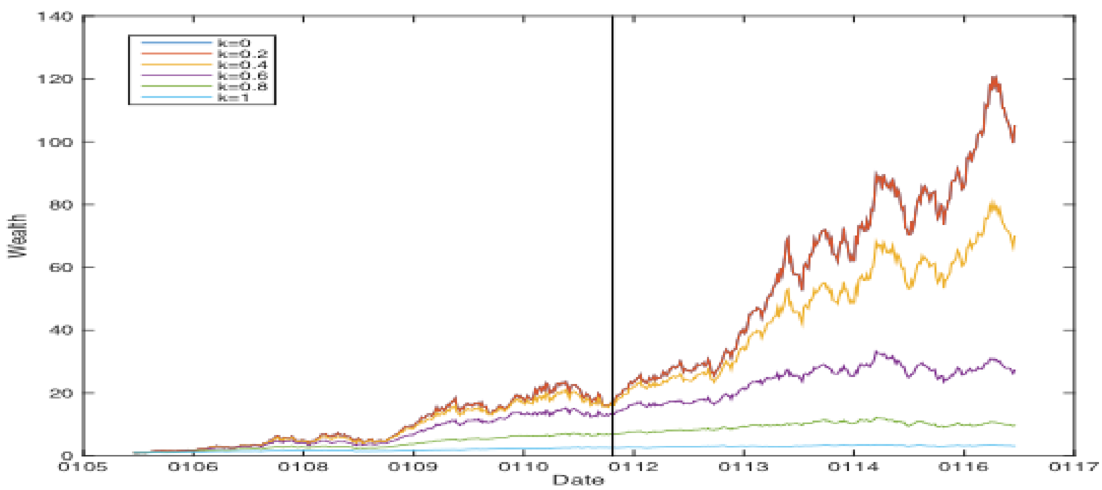


Figure 4.6 is the wealth of the Markowitz model based on different of *k*. The left side is the in-sample data and right side is the out-of-sample data. We can see there’s a significantly

rising in the wealth of the portfolio in the out-of-sample period. The highest wealth is when k=0.2. The highest wealth is 120.860. The wealth of the strategy is from 1 to 121. As value of k becomes lower, we get lower wealth line. The strategy with the lowest wealth is situation when k=1. For this strategy, the highest wealth is 3.608.

Table 4.6 In-sample result of Markowitz model

k values	0.2	0.4	0.6	0.8	1
Final wealth	16.59	16.28	13.44	6.84	2.64
Return	1558.96%	1527.20%	1221.82%	583.14%	164.22%
Annual Return	59.94%	59.42%	53.98%	37.89%	17.64%
Std. (annual)	46.93%	39.77%	30.56%	19.41%	13.77%
Maximum drawdown	44.25%	36.02%	33.19%	25.66%	25.75%
Sharpe ratio	16.29%	18.04%	20.51%	22.45%	14.68%

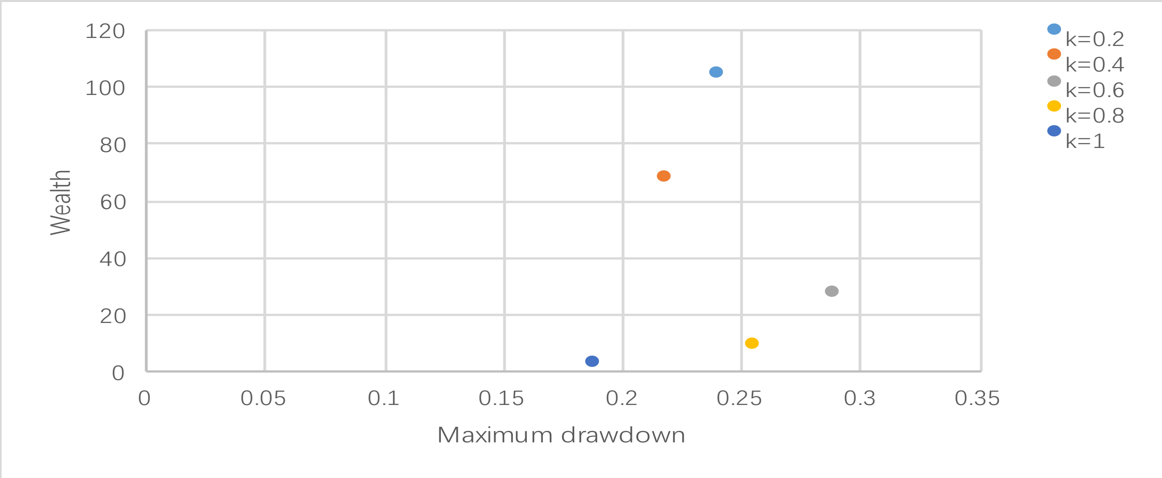
Table 4.7 Out-of-sample result of Markowitz model

k values	0.2	0.4	0.6	0.8	1
Final wealth	105.07	68.62	27.86	9.76	3.14
Return	539.93%	333.86%	105.95%	40.60%	17.73%
Annual Return	44.75%	33.96%	15.48%	7.03%	3.31%
Std. (annual)	27.47%	23.90%	19.61%	15.13%	12.60%
Maximum drawdown	24.06%	21.80%	28.92%	25.61%	18.84%
Sharpe ratio	19.27%	17.09%	9.60%	4.73%	1.39%

Based on the results in table 4.5, we can calculate the in-sample and out-of-sample performance measures of the portfolio. Table 4.6 is the in-sample results and table 4.7 is the out-of-sample results. We can see that when the k values increase, there are increase in both in sample and out-of-sample wealth and increase in standard deviation, which means if the receive higher return, we need to accept higher risk.

For performance ratios in out-of-sample period, the highest Sharpe ratio occurred when k=0.2. In the meanwhile we reach the highest wealth. The rank of maximum drawdown for k=0.2 is 3rd, which is acceptable. When k=1, we get the lowest risk and when k=0.2, we reach highest wealth. Moreover, we can compare the performance measures and the wealth. Figure 4.7 shows the maximum drawdown and wealth. We prefer higher wealth and lower maximum drawdown. When k=1, we get the lowest maximum drawdown. However, the wealth is in a very low level. When k=0.4, the maximum drawdown is in the second level and the wealth is also second level. For k=0.2, we reach the highest wealth and the maximum drawdown is in the acceptable level.

Figure 4.7 Maximum drawdown of Markowitz model



From figure 4.7 and table 4.7 we can see that when k=0.2 and k=0.4, when k level increase, the maximum drawdown decreased. For other k levels, as k increase the maximum drawdown is lower. The increase of k level also increases the return and decreases the Sharpe ratio and standard deviation. Based on maximum drawdown we can conclude that when k=0.2, k=0.4 and k=1, the strategies can be accepted because the results are not dominated.

4.4.2 Backtesting of Markowitz Model based on Rolling Window strategy

In this section we apply the rolling window strategy to the out-of-sample data. Instead of simply calculate weight from the in-sample data, we use the weight based on the latest data. We assume that we start investing in 312th week. The initial wealth is 1 HKD. In the 312th week, we use the weight based on the 311 weeks before and in week 313, we use the data from the 2nd week to 312th week to calculate the weight and invest. We also apply different k level and make comparison of simple approach. By running program A5, we can get the result in Table 4.8.

Table 4.8 Result of Markowitz model

Dates	k=0		k=0.2		k=0.4		k=0.6		k=0.8		k=1	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2012/1/9	6.82%	1.10	0.07	1.10	0.05	1.07	0.04	1.04	0.01	1.01	-0.02	0.98
…	…	…	…	…	…	…	…	…	…	…	…	…
2012/8/6	6.62%	1.53	0.07	1.53	0.06	1.42	0.04	1.20	0.02	1.11	0.00	1.04
2012/8/13	-1.29%	1.63	-0.01	1.63	-0.01	1.50	0.00	1.24	0.01	1.14	0.01	1.04
…	…	…	…	…	…	…	…	…	…	…	…	…
2013/12/9	-1.79%	3.46	-0.01	3.32	-0.01	3.03	0.00	2.03	0.01	1.44	0.00	1.10
2013/12/16	3.56%	3.40	0.04	3.27	0.04	3.00	0.03	2.02	0.02	1.46	0.02	1.11
…	…	…	…	…	…	…	…	…	…	…	…	…
2014/8/18	-4.81%	3.16	-0.05	3.05	-0.04	3.40	-0.04	2.33	-0.04	1.71	-0.03	1.30
…	…	…	…	…	…	…	…	…	…	…	…	…
2016/3/28	-0.89%	1.44	-0.01	1.39	0.00	1.73	0.00	1.68	0.01	1.50	0.00	1.31
2016/4/4	3.78%	1.43	0.04	1.38	0.04	1.72	0.03	1.68	0.02	1.52	0.03	1.31
…	…	…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-0.02	1.18	-0.02	1.28	-0.02	1.75	-0.02	1.82	-0.02	1.51	-0.02	1.41
2016/12/26	0.06	1.16	0.06	1.25	0.05	1.71	0.05	1.78	0.03	1.48	0.00	1.38

Table 4.8 shows the results of Markowitz model. We apply 6 different *k* levels. The highest final wealth is when *k*=0.6. The lowest wealth is when *k*=0. The final wealth of all strategies is between 1 and 2 HKD.

Figure 4.8 Comparison of the portfolio returns

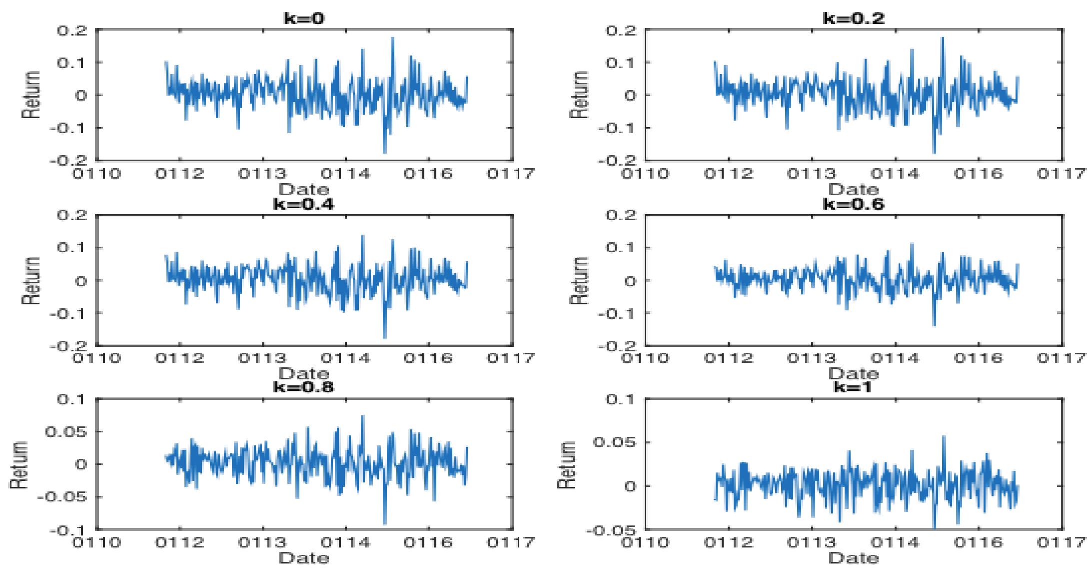


Figure 4.8 is the comparison of the portfolio returns. We can see the returns of the portfolio are between -0.2 and 0.2 HKD. When *k*=0, *k*=0.2 and *k*=0.4, the returns have similar trend. Nearly half of the returns are negative. When *k*=1, the returns are in smaller range.

Figure 4.9 Wealth of Markowitz model

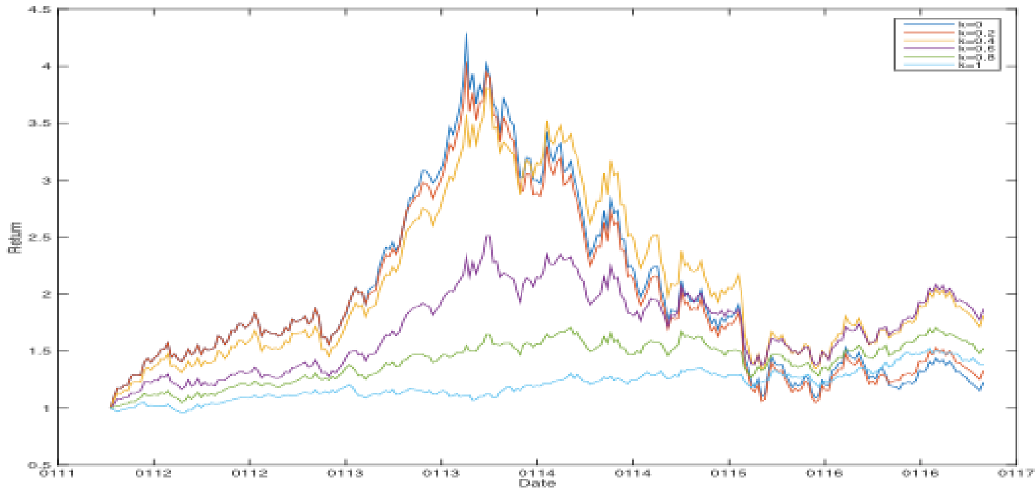


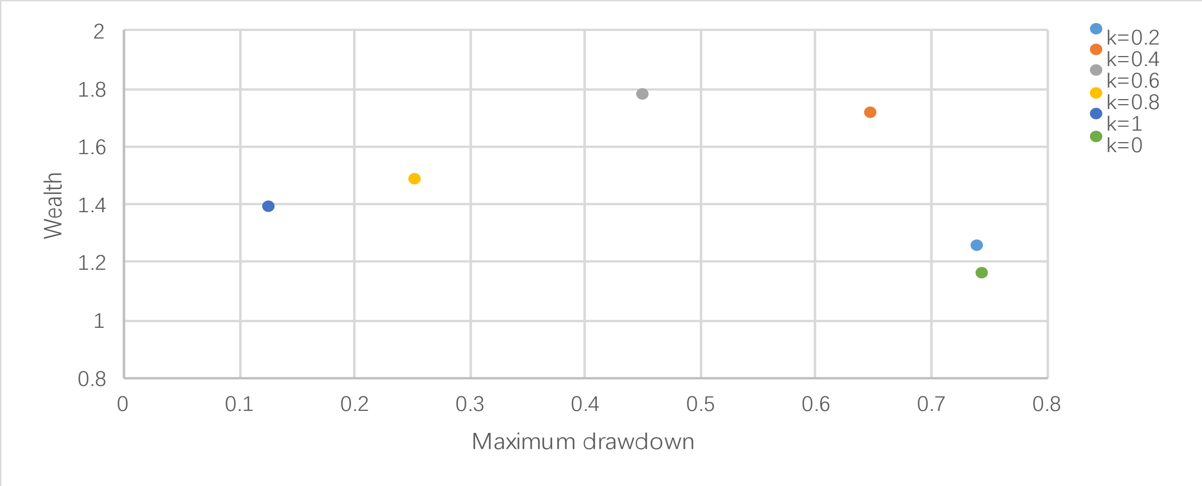
Figure 4.9 represent the wealth of the Markowitz model. We can see significant increase in the rolling approach from 2013 to 2014. The highest wealth among all strategies increases for nearly 5 times in this period. Then the wealth had a decrease trend from 2014 to 2015. Because there's a decrease in house price, people are more willing to invest in stock market. The companies will get more cash flow for operating, which may lead to increase in all prices. However the stock price decrease due to some political reasons. We get lower return after that period.

Table 4.9 Results of Markowitz model

k values	0	0.2	0.4	0.6	0.8	1
Final wealth	1.156	1.252	1.712	1.776	1.483	1.384
Return	22.00%	32.19%	80.64%	86.26%	52.04%	38.43%
Annual Return	4.04%	5.72%	12.50%	13.19%	8.71%	6.69%
Std. (annual)	35.15%	34.56%	31.51%	24.63%	16.45%	12.09%
Maximum drawdown	74.58%	73.97%	64.80%	45.15%	25.43%	12.65%
Sharpe ratio	2.90%	3.51%	6.17%	7.14%	5.85%	5.09%

We present the result of the model in table 4.9. We can see that the return for rolling window approach, we get highest return rate when *k*=0.6 and the highest final wealth, which is 1.776. The annual return in this situation is also the highest. As for performance measures, the highest Sharpe ration is 7.14%, when *k*=0.6. The maximum drawdown in this situation is 45.15%. The lowest standard deviation is when *k*=1, which is 12.09%. When *k*=0, the wealth and Sharpe ratio is in a minimum number. The second choice is when *k*=0.4, we got acceptable Sharpe ratio and wealth level.

Figure 4.10 Maximum drawdown of Markowitz model



From figure 4.10 and table 4.9 we can see that for chosen strategies, as the k level increase, the maximum drawdown decrease. The Sharpe ratio and return first increase (from k=0 to k=0.6) and then decrease (from k=0.8 to k=1). The acceptable strategies are k=0.6, k=0.8 and k=1.

4.5 Portfolio Optimization of CVaR Model

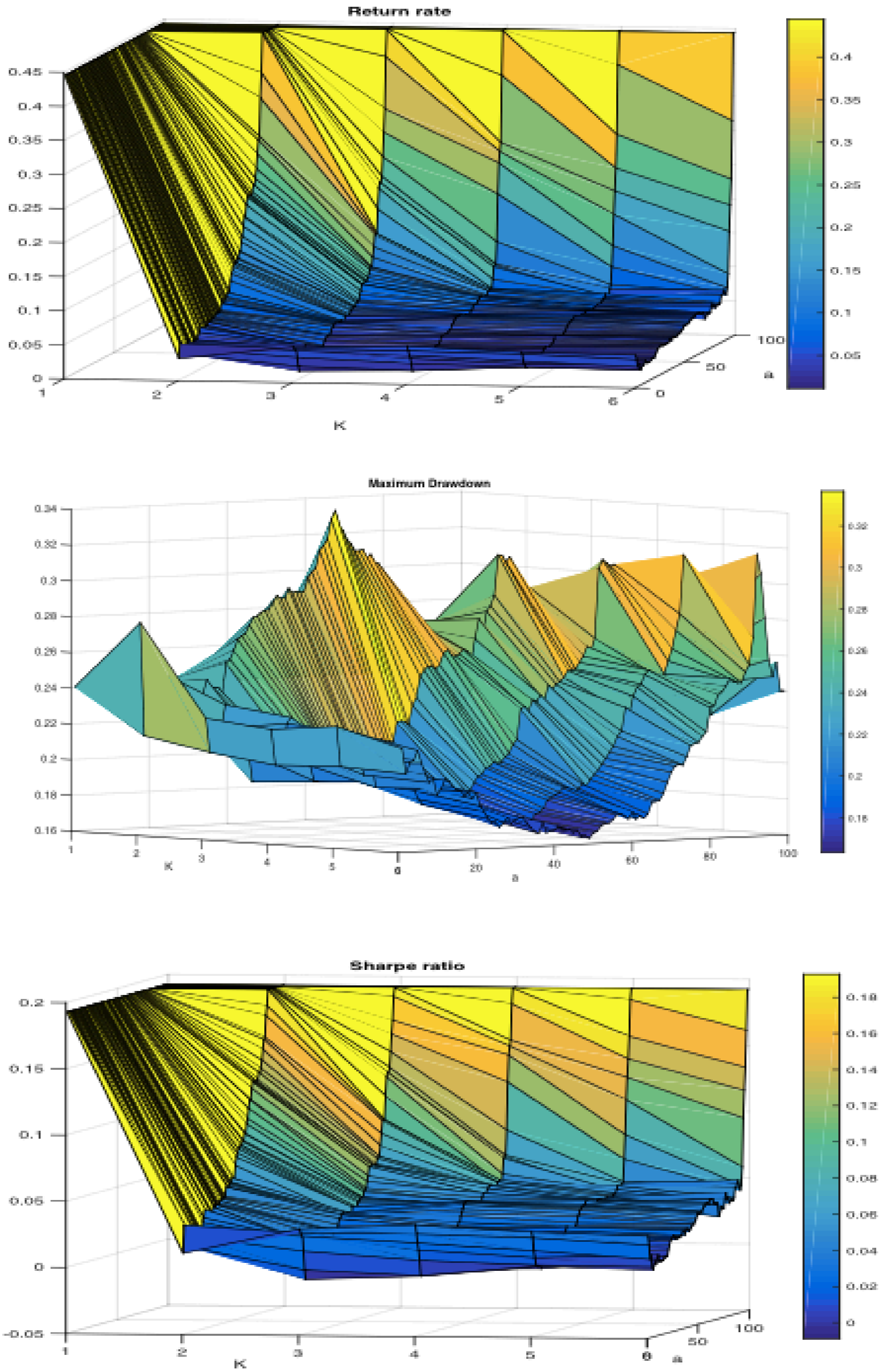
Conditional value at risk is a risk measurement method developed on the basis of VaR. In this chapter we apply CVaR model to our portfolio and use different k and α levels and use the rolling window principle for backtesting.

4.5.1 Backtesting of CVaR Model with Simple Approach

Firstly based on simple approach we calculate different k and α values. We calculate the combination of k values equals to 0, 0.2, 0.4, 0.6, 0.8 and 1 and the chosen α levels (1%, 2%, ..., 99%) and also the Sharpe ratio and maximum drawdown and see the result of the performance. For different combination we get different result. By running the program A6 in Annex, we can get the result as figure 4.14.

In figure 4.11 we apply different α level and k. The x axis is the α level (%). For y axis, the variable $k=\frac{(K-1)}{5}$, which represents different k level. We use 'surf ' in Matlab to plot to create a three-dimensional surface plot. We use the function to plot the return rate, maximum drawdown and Sharpe ratio as heights above a grid in the K- α plane. The height is expressed as color data, so the color is proportional to height.

Figure 4.11 Analysis of parameter k and α on return rate (top), maximum drawdown (middle) and Sharpe ratio (bottom)



The goal is to choose higher level of return and Sharpe ratio and lower maximum drawdown. The areas with yellow color for Sharpe ratio and return rate represents better result. For maximum drawdown, we need to choose dark blue areas.For higher return level, we prefer the combination of lower *k* level and α level lower than 85% or α higher than 85% combined with higher *k*. The situation is similar for Sharpe ratio. For maximum drawdown, we need to combine high level of *k* and middle level of α to get minimum result. To show the result more clearly, in the next section we use some chosen α and *k* levels and calculate the results.

4.5.1.1 Backtesting of CVaR Model with Different *k*

Firstly, we apply different *k* values for CVaR model. We set the α level at 0.15 and use different *k* level. Also we do the same thing as Markowitz model. We divided our data into two parts. By running the codes in Program A7, we get result in table 4.10.

Table 4.10 Wealth and return of CVaR model based on different *k*

Dates	k=0.2		k=0.4		k=0.6		k=0.8		k=1	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2006/1/23	0.0091	1.0091	0.0128	1.0128	0.0145	1.0145	0.0149	1.0149	0.0142	1.0142
…	…	…	…	…	…	…	…	…	…	…
2007/6/11	0.0133	1.8746	0.0096	1.6028	0.0111	1.5824	0.0093	1.5388	0.0085	1.5182
2007/6/18	-0.0036	1.8677	-0.0068	1.5919	-0.0057	1.5733	-0.0063	1.5291	-0.0075	1.5069
…	…	…	…	…	…	…	…	…	…	…
2011/12/19	0.0107	4.6956	0.0043	3.2957	0.0040	3.0880	0.0029	2.9139	0.0024	2.8380
2011/12/26	-0.0044	4.6747	-0.0107	3.2603	-0.0099	3.0574	-0.0099	2.8852	-0.0098	2.8102
2012/1/2	0.0074	4.7091	0.0114	3.2974	0.0119	3.0938	0.0121	2.9201	0.0123	2.8448
2012/1/9	-0.0037	4.6916	-0.0160	3.2447	-0.0184	3.0370	-0.0169	2.8708	-0.0166	2.7976
…	…	…	…	…	…	…	…	…	…	…
2016/6/13	-0.0546	5.5957	-0.0566	3.7258	-0.0560	3.3530	-0.0563	3.2130	-0.0561	3.1407
2016/6/20	0.0012	5.6023	0.0005	3.7275	0.0004	3.3544	0.0004	3.2142	0.0002	3.1414
…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-0.0202	5.7803	-0.0179	3.8402	-0.0173	3.4585	-0.0162	3.3090	-0.0158	3.2259
2016/12/26	0.0114	5.8462	0.0044	3.8572	0.0028	3.4682	0.0018	3.3148	0.0014	3.2304

Table 4.10 shows the wealth and returns of the CVaR model of different *k* levels. From 2006 to 2011 are in-sample data and from 2012 to 2016, data is out-of-sample. The initial wealth is 1 HKD and we set the α level at 0.15. The chosen *k* level is 0.2, 0.4, 0.6, 0.8 and 1. As the situation when *k*=0 is the same as Markowitz model because in this situation we didn't consider the risk, we skip this *k* level. The upper side is the in-sample data and lower side is the out-of-sample data. From table 4.10 we can see the highest in-sample return is at 10th March 2008 when *k*=0.8. The highest return is 0.094. The lowest return is 6th October 2008,

when k=1. For out-of-sample period, the highest return is when k=0.2 at 6th April 2015, with the value of 0.048. The lowest return is 0.056, when k=0.4 at 13th June 2016. Furthermore, we can plot the wealth to have better view of the situation. Figure 4.12 Returns of CVaR model based on different k levels.

Figure 4.12 Returns of CVaR model based on different k levels

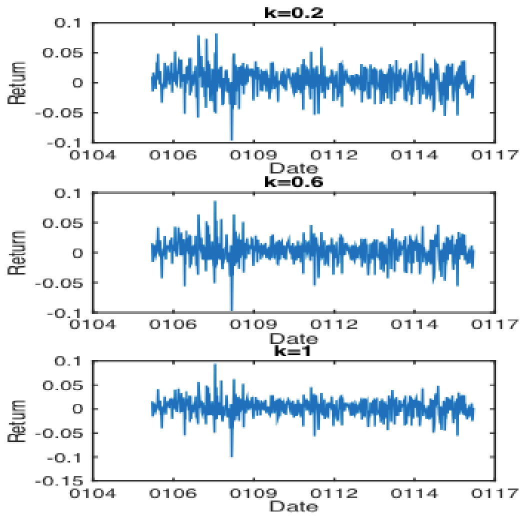
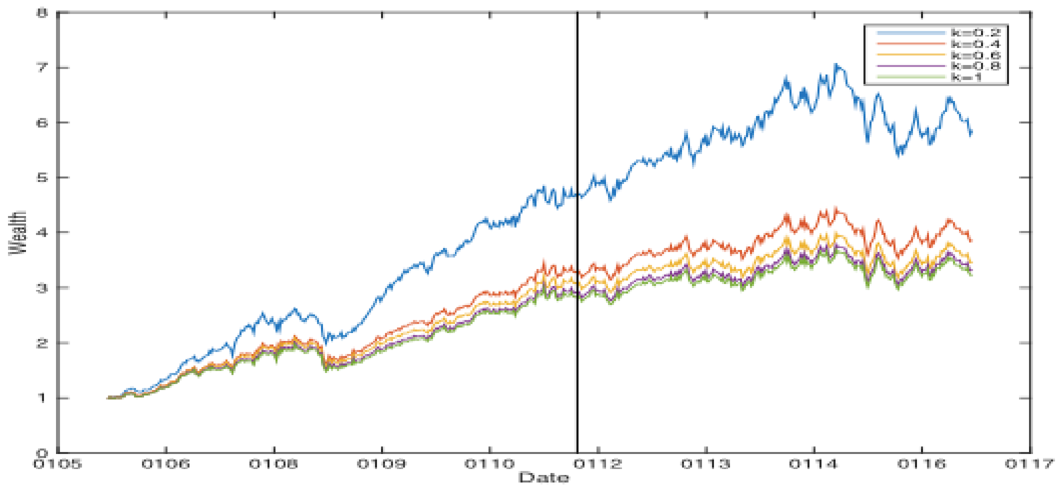


Figure 4.12 is the returns of the model. We can see that when k=0, we have higher range of return. Moreover, all of these 5 strategies all have decreased and reach minimum return at 2008 for all k level, which may cause by financial crisis. In out-of-sample period, the returns for portfolio are stable.

Figure 4.13 Wealth of CVaR model based on different k values



From figure 4.13 we can say that with lower *k* level, the wealth of the portfolio increase. When *k*=0.2, the wealth level is significantly higher than other strategies. For in-sample data, there's a decrease in 2008, especially in June. Because there's decrease in nearly all stocks price, no matter what decision we decide to invest, we have negative return. Also there's a decrease in 2010 caused by devaluation of HKD. The stock price decreased that we receive loss.

Table 4.11 In-sample Results of CVaR model of different *k* values

k values	0.2	0.4	0.6	0.8	1
Final wealth	4.67	3.26	3.06	2.89	2.81
Return	369.56%	229.57%	208.80%	191.39%	183.80%
Annual Return	29.51%	22.07%	20.75%	19.58%	19.05%
Std. (annual)	16.02%	14.44%	14.30%	14.27%	14.25%
Maximum drawdown	24.09%	23.61%	24.56%	24.73%	24.89%
Sharpe ratio	21.16%	17.51%	16.62%	15.70%	15.30%

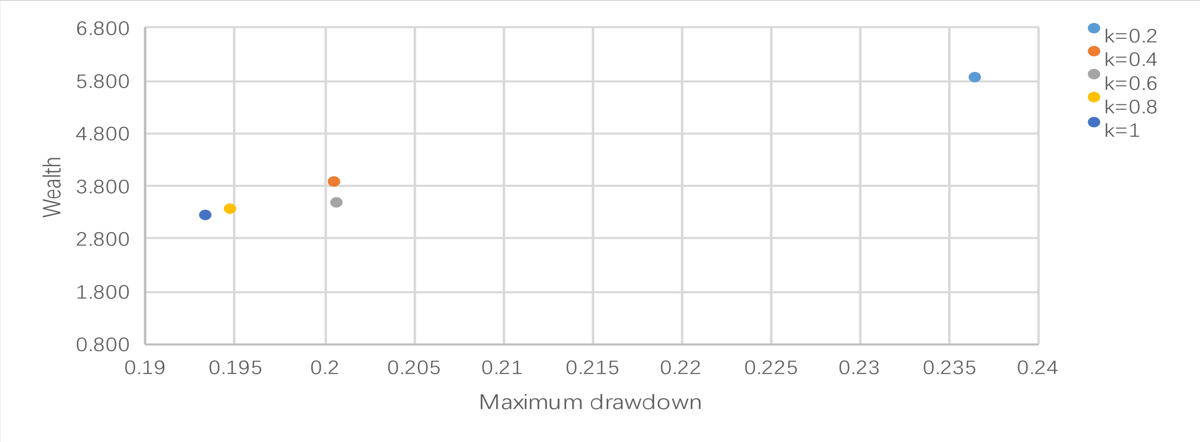
Table 4.11 shows the in-sample result of CVaR model. As *k* increase, the final wealth has a decrease tendency for chosen *k* levels. The return rate and the annual return are with the same tendency. For standard deviation and Sharpe ratio, as *k* increase the ratio decrease. The lowest maximum drawdown is when *k*=0.4. The result is in very similar level.

Table 4.12 Out-of-sample Results of CVaR model of different *k* values

k values	0.2	0.4	0.6	0.8	1
Final wealth	5.85	3.86	3.47	3.31	3.23
Return	25.06%	18.31%	13.44%	14.89%	14.95%
Annual Return	4.56%	3.41%	2.54%	2.80%	2.81%
Std. (annual)	13.74%	13.11%	13.04%	12.97%	12.92%
Maximum drawdown	23.65%	20.07%	20.07%	19.48%	19.34%
Sharpe ratio	2.49%	1.27%	0.36%	0.62%	0.62%

Table 4.12 is the out-of-sample result for CVaR model. The final wealth, return and annual return decrease when increase *k* level. The highest wealth is then *k*=0.2. The highest Sharpe ratio occurs when *k* =0.2, which is 2.49%. The minimum maximum drawdown is when *k* =1. The value is 19.34%. We plot the final wealth and performance measures in figure 4.14. The strategy when *k* =0.2 is with the highest maximum drawdown. When *k* =0.4, we get second rank of Sharpe ratio and 3rd level of maximum drawdown.

Figure 4.14 Wealth and performance measures of CVaR model of different *k*



From table 4.12 and figure 4.14 we get that as *k* level increases, the wealth and Sharpe ratio decreases. The maximum drawdown also decreases. Based on maximum drawdown, the strategy when *k*=0.6 is dominated. There exist better strategies. The other strategies are not dominated.

4.5.1.2 Backtesting of CVaR Model with Different α Level

In this section we discuss about the strategy of different α level. We set the *k*=0.2 and evaluate the impact of α will cause on the portfolios. Using program A8, we can get the result as table 4.13.

Table 4.13 Result of CVaR model of different α level

Dates	α =0.005		α =0.01		α =0.05		α =0.1		α =0.15	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2006/1/23	-0.010	0.990	-0.004	0.996	0.008	1.008	0.012	1.012	0.009	1.009
…	…	…	…	…	…	…	…	…	…	…
2007/3/19	0.027	2.004	0.021	1.686	0.027	1.494	0.024	1.540	0.026	1.700
2007/3/26	-0.016	1.973	-0.012	1.665	0.007	1.505	0.004	1.546	-0.002	1.696
…	…	…	…	…	…	…	…	…	…	…
2011/12/19	0.014	6.653	0.009	5.215	0.013	4.036	0.008	3.936	0.011	4.696
2011/12/26	-0.033	6.430	-0.015	5.138	-0.003	4.023	-0.004	3.919	-0.004	4.675
2012/1/2	0.010	6.492	0.012	5.199	0.004	4.040	0.007	3.948	0.007	4.709
2012/1/9	-0.013	6.404	-0.005	5.174	-0.016	3.976	-0.009	3.911	-0.004	4.692
…	…	…	…	…	…	…	…	…	…	…
2013/5/13	0.025	8.008	0.028	6.552	0.017	5.117	0.019	4.823	0.024	5.799
2013/5/20	-0.009	7.938	-0.016	6.447	-0.028	4.973	-0.016	4.747	-0.018	5.693
…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-0.033	6.685	-0.024	5.920	-0.019	4.918	-0.017	4.967	-0.020	5.780
2016/12/26	0.024	6.846	0.016	6.013	0.013	4.980	0.008	5.007	0.011	5.846

Table 4.13 shows the result of CVaR model based on different α level. We can see that as the α level increase the width of the data selected will increase. The highest wealth of all strategies in in-sample period is on 28th November 2011 at α=0.005. The highest wealth is 6.669.

Figure 4.15 Return of CVaR model of different α level

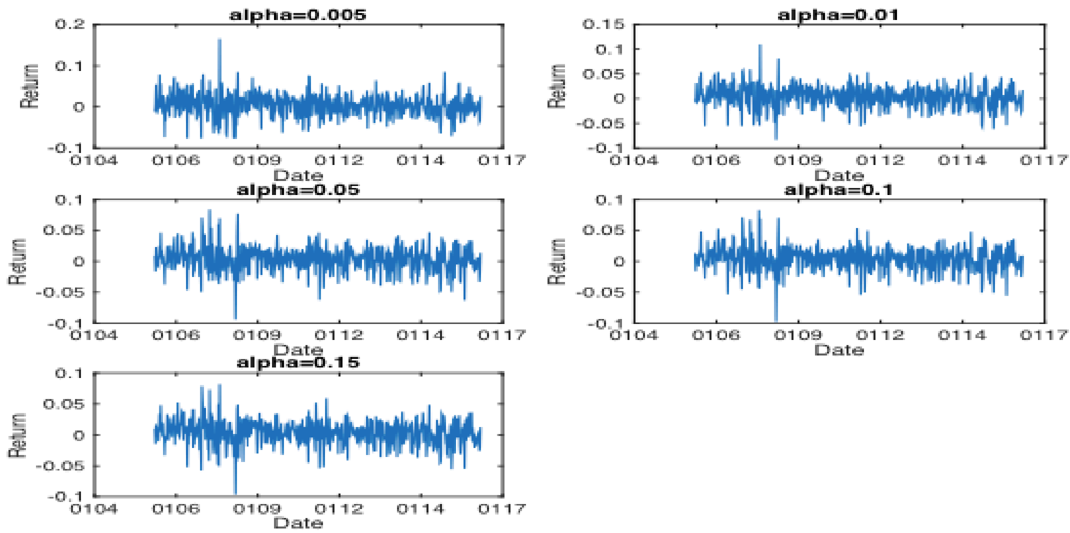


Figure 4.15 is the returns of the CVaR model of different α level. We get similar level of portfolio returns. The range of returns is from -0.1 to 0.1. All of the strategies have a loss of portfolio at 2008, which may because of the financial crisis.

Figure 4.16 Wealth of CVaR model of different α level

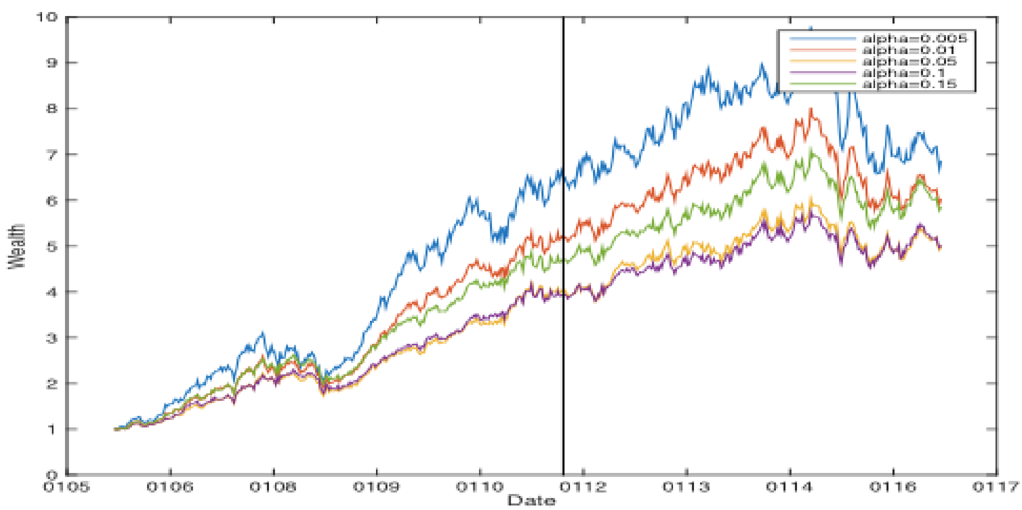


Figure 4.16 shows portfolio wealth change of different α level. We can see all of the wealth have the similar tendency. All of the strategies suffer a loss in 2008 and 2015. The final wealth of chosen strategies is from 5 to 6. The highest result is when α=0.005. When α=0.01, we also have high portfolio wealth. The lowest level of final wealth is when α=0.15. Basically for the chosen α level we can say that as α decrease, the wealth of the portfolio will increase.

Table 4.14 In-sample result of CVaR model of different α level

α	α =0.005	α =0.01	α =0.05	α =0.1	α =0.15
Final wealth	6.43	5.14	4.02	3.92	4.68
Return	565.31%	421.46%	303.55%	293.57%	369.56%
Annual Return	37.28%	31.80%	26.27%	25.74%	29.51%
Std. (annual)	23.57%	17.88%	15.73%	15.45%	16.02%
Maximum drawdown	34.09%	26.17%	22.40%	22.05%	24.09%
Sharpe ratio	18.43%	20.50%	19.24%	19.21%	21.16%

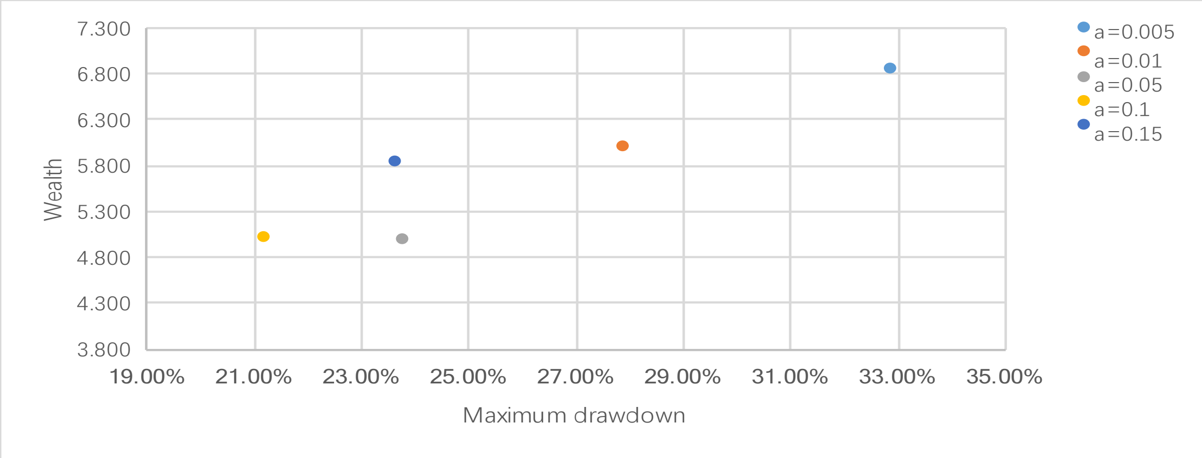
From table 4.14 we can see the in-sample results of the CVaR model. When α=0.005, we have the highest in-sample return, weekly and annual return. The standard deviation is also high. For the chosen data as α level increase, the returns will decrease the standard deviation decrease. The highest Sharpe ratio is when α = 0.15.

Table 4.15 Out-of-sample result of CVaR model of different α level

α	α =0.005	α =0.01	α =0.05	α =0.1	α =0.15
Final wealth	6.85	6.01	4.98	5.01	5.85
Return	6.47%	17.03%	23.80%	27.78%	25.06%
Annual Return	1.26%	3.18%	4.34%	5.00%	4.56%
Std. (annual)	17.22%	14.82%	14.27%	13.42%	13.74%
Maximum drawdown	32.86%	27.89%	23.79%	21.20%	23.65%
Sharpe ratio	-0.21%	1.13%	2.33%	2.95%	2.49%

From table 4.15 we see the out-of-sample result of chosen strategy. As α level from 0.0005 to 0.05 there's a decrease in final wealth, returns standard deviation and maximum drawdown. The Sharpe ratio is increasing in this situations. We plot the result in figure 4.20. When α=0.005, we get negative Sharpe ratio. The wealth of this strategy is the highest level. The highest maximum drawdown is α=0.005 and the lowest level is α =0.1. The highest Sharpe ratio is when α=0.1 and the second rank is α=0.15.

Figure 4.17 Wealth and performance measures of CVaR model



Based on figure 4.17 and table 4.15 we can conclude that from α=0.005 to α=0.1, the Sharpe ratio and return rate increased. The annual standard deviation decreased. There's no significant trend for maximum drawdown. Except for α=0.05, all strategies are not dominated.

4.5.2 Backtesting of CVaR Model with Rolling Window Principle

In this section we apply the rolling window strategy to our model. The section is divided into 2 parts. In the first part we compare the strategy of different k values by setting the α level at 0.15 and calculate the out-of-sample result. In the second part we apply different α values. The chosen k level is 0.2.

4.5.2.1 Rolling Window Strategy of Different *k* Levels

In this section we apply rolling window strategy on different k levels. We set α=0.15 and invest into portfolio with different strategy every week to evaluate results. We run the program A9 and get the results.

Table 4.16 is the result after running program A9. We get the result as table 4.16. All strategy has final wealth more than initial wealth. The second wealth is obtained by k =1. The highest wealth is in 5th September 2016 when k =0.2. The lowest portfolio return is also happens when k =0.2. The minimum return is -0.073. For more than half of the weeks we get negative returns. When k level increase, there's no significant trend of the change in final wealth. At the final week, our wealth increased by 0.3057 HKD. We can plot the result of wealth in figure 4.18 to present the change of wealth through time.

Table 4.16 Wealth and return of CVaR model

Dates	k=0.2		k=0.4		k=0.6		k=0.8		k=1	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2012/1/9	-0.0042	0.9958	-0.0164	0.9836	-0.0185	0.9815	-0.0171	0.9829	-0.0169	0.9831
…	…	…	…	…	…	…	…	…	…	…
2012/8/13	0.0068	1.0429	-0.0044	1.0174	-0.0040	1.0152	-0.0030	1.0231	-0.0033	1.0267
…	…	…	…	…	…	…	…	…	…	…
2013/6/3	-0.0247	1.1645	-0.0184	1.0945	-0.0181	1.0873	-0.0180	1.1008	-0.0181	1.1001
…	…	…	…	…	…	…	…	…	…	…
2013/12/16	0.0014	1.2107	0.0053	1.0719	0.0055	1.0630	0.0060	1.0763	0.0059	1.0788
…	…	…	…	…	…	…	…	…	…	…
2016/4/11	0.0235	1.3400	0.0245	1.2318	0.0244	1.2222	0.0251	1.2715	0.0249	1.3068
2016/4/18	-0.0109	1.3253	-0.0022	1.2291	0.0018	1.2244	0.0045	1.2772	0.0048	1.3131
…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-0.0169	1.3264	-0.0166	1.2291	-0.0170	1.2126	-0.0166	1.2668	-0.0165	1.3012
2016/12/26	0.0146	1.3457	0.0098	1.2411	0.0039	1.2173	0.0036	1.2714	0.0035	1.3057

Figure 4.18 Wealth of CVaR model

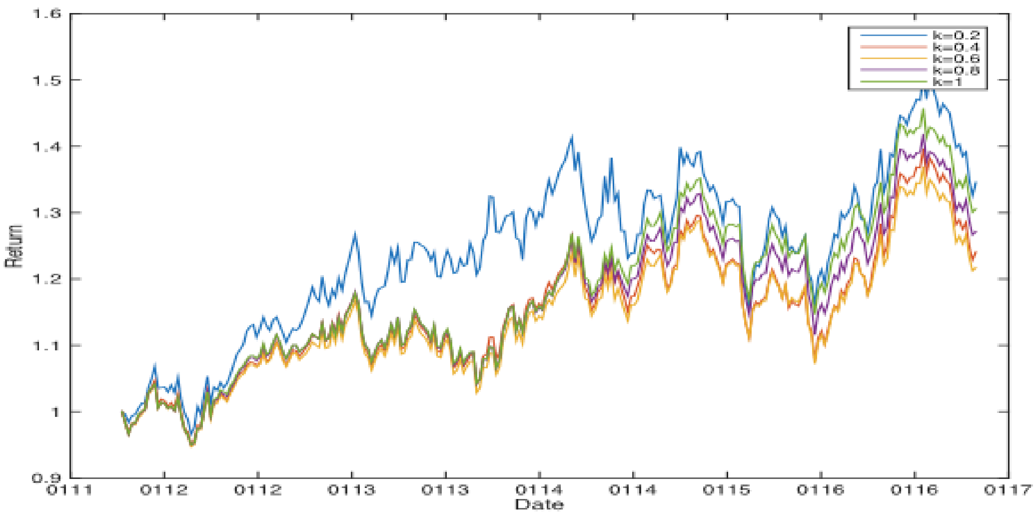


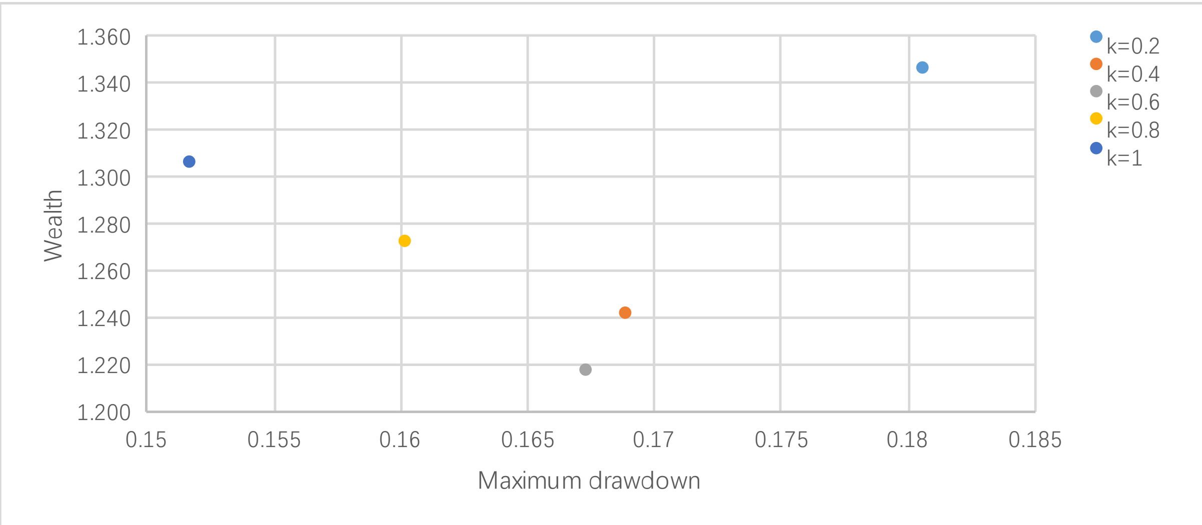
Figure 4.18 shows the change of wealth of our strategy. We can see that from 2011 to 2015, our wealth is increasing. Then the wealth has a decreasing tendency and decreased to nearly 1.1. Then the wealth increase to its peak in 2016 and after that it has a decreasing tendency. The wealth when k =0.2 is higher than other strategy in most periods.

Table 4.17 Result of CVaR model

k values	0.2	0.4	0.6	0.8	1
Final wealth	1.35	1.24	1.22	1.27	1.31
Return	34.57%	24.11%	21.73%	27.14%	30.57%
Annual Return	6.09%	4.40%	4.00%	4.90%	5.46%
Std. (annual)	14.10%	12.35%	12.17%	12.04%	11.96%
Maximum drawdown	18.06%	16.89%	16.73%	16.02%	15.17%
Sharpe ratio	4.07%	2.56%	2.13%	3.14%	3.77%

Table 4.17 is the result of CVaR model of the rolling window approach based on different k level. We can see that for rolling window approach, we get earnings for the final wealth. The highest final wealth is when k =0.2. The second rank of return we get from k =1. The maximum drawdown decreases as k values increases. The highest Sharpe ratio is when k =0.2, which is the highest point in lower figure. The second rank of Sharpe ratio is k =1, where the wealth and Sharpe ratio are all second rank. The maximum drawdowns are in similar level. The lowest maximum drawdowns is when k =1.

Figure 4.19 Wealth and performance measures of CVaR model



From figure 4.19 and table 4.17, from k=0.2 to k=0.6, the return and Sharpe ratio increased. For k=0.8 and k=1, the return and Sharpe ratio decreased. For all chosen k levels, as k increase, standard deviation and maximum drawdown decreased. Only when k=0.2 and k=1 the strategy is not dominated.

4.5.2.2 Rolling Window Strategy of Different α Levels

In this section we apply rolling window strategy on different α levels. The chosen k level is 0.2. The initial wealth is 1 HKD. We run program A10 and the return and wealth is in table 4.18.

Table 4.18. Wealth and return of rolling window strategy

Dates	α =0.005		α =0.01		α =0.05		α =0.1		α =0.15	
	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth	Return	Wealth
2012/1/9	-0.0145	0.9855	-0.0034	0.9966	-0.0152	0.9848	-0.0092	0.9908	-0.0042	0.9958
…	…	…	…	…	…	…	…	…	…	…
2012/5/28	-0.0360	0.9922	-0.0305	0.9944	-0.0293	0.9414	-0.0224	0.9562	-0.0218	0.9667
2012/6/4	0.0103	1.0024	0.0088	1.0032	0.0079	0.9489	0.0101	0.9659	0.0101	0.9764
…	…	…	…	…	…	…	…	…	…	…
2013/4/15	-0.0182	1.1422	-0.0167	1.1813	-0.0134	1.1611	-0.0168	1.1811	-0.0147	1.1837
2013/4/22	0.0255	1.1714	0.0262	1.2122	0.0269	1.1923	0.0216	1.2066	0.0215	1.2091
…	…	…	…	…	…	…	…	…	…	…
2014/3/10	-0.0319	1.2563	-0.0355	1.2188	-0.0361	1.2113	-0.0364	1.2531	-0.0395	1.2710
…	…	…	…	…	…	…	…	…	…	…
2016/3/7	-0.0027	1.1187	-0.0027	1.1638	-0.0021	1.1337	-0.0058	1.1983	-0.0039	1.2626
2016/3/14	0.0276	1.1496	0.0269	1.1950	0.0218	1.1584	0.0280	1.2318	0.0299	1.3004
…	…	…	…	…	…	…	…	…	…	…
2016/12/19	-0.0175	1.1754	-0.0183	1.2195	-0.0186	1.2319	-0.0177	1.2768	-0.0169	1.3264
2016/12/26	0.0087	1.1856	0.0055	1.2262	0.0096	1.2437	0.0148	1.2956	0.0146	1.3457

For the chosen strategy we get relatively steady result of wealth. The lowest wealth is at 28th May 2012, which is 0.9414. We only get small loss. In most of period the wealth for chosen strategy is higher than initial level. The highest wealth and lowest return are all in the strategy α=0.15. The highest wealth is 1.504 HKD. In 6th April 2015 we get highest return of chosen strategy when α=0.005. The highest portfolio return is 0.068.

Figure 4.20 Wealth of CVaR model

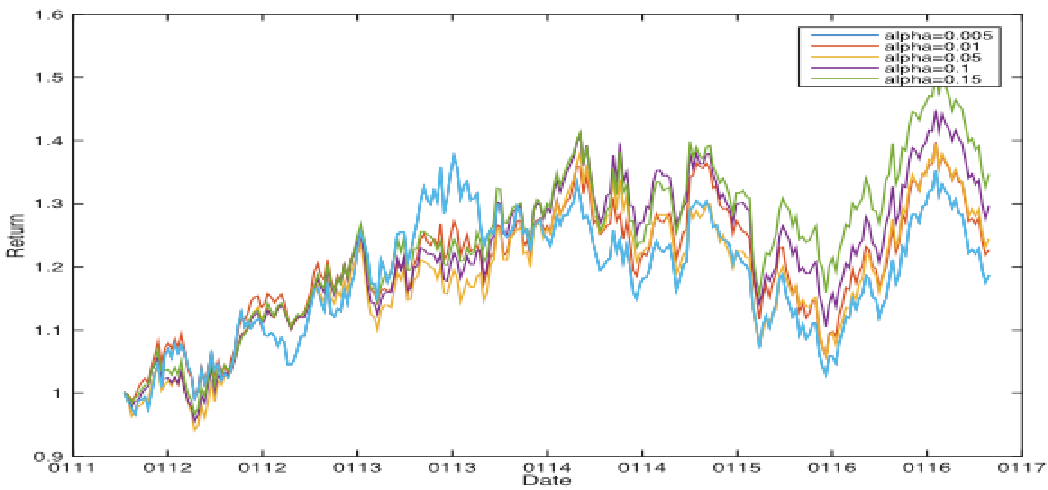


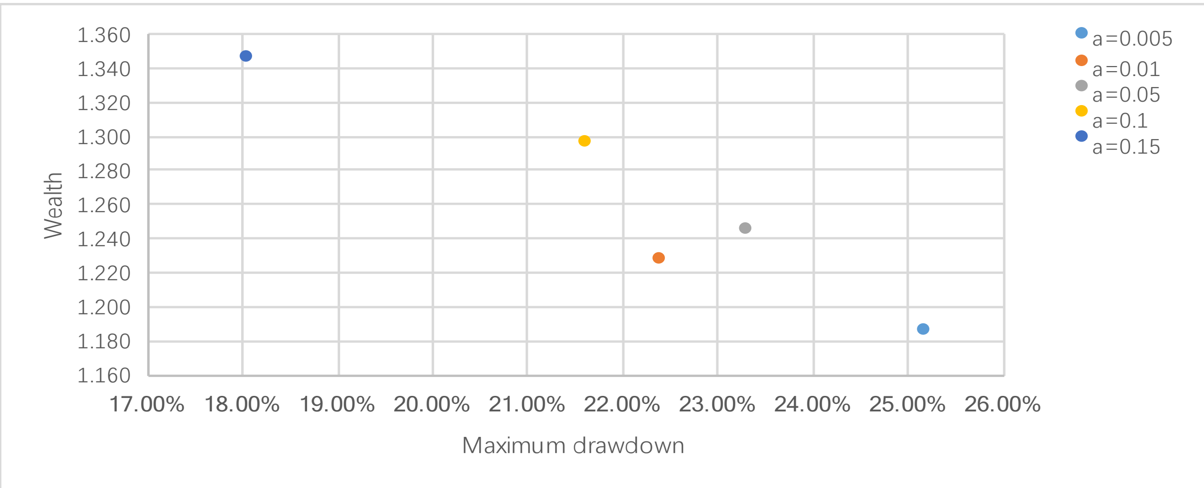
Figure 4.20 is the plot of wealth of chosen strategy. The wealth has no clearly relationship with α levels. The final wealth is higher as the α increased. The wealth has relatively steady increase from 2012 to 2015. In 2016, the wealth decreased to a lower level and increase sharply after that. At the end the wealth decreased. The final wealth of all α is higher than initial wealth.

Table 4.19 Result of CVaR model

α	α =0.005	α =0.01	α =0.05	α =0.1	α =0.15
Final wealth	1.19	1.23	1.24	1.30	1.35
Return	18.56%	22.62%	24.37%	29.56%	34.57%
Annual Return	3.45%	4.15%	4.44%	5.30%	6.09%
Std. (annual)	14.74%	13.65%	14.00%	14.02%	14.10%
Maximum drawdown	25.16%	22.40%	23.31%	21.61%	18.06%
Sharpe ratio	1.59%	2.24%	2.52%	3.33%	4.07%

The result of CVaR model using rolling window strategy of different α is in table 4.19. For chosen α levels, the final wealth and returns increase. The highest standard deviation is when α=0.005. The second rank is α=0.15. The Sharpe ratio of α=0.15 is significantly higher than other strategies. Also in this situation we get lowest maximum drawdown. The second rank is when α=0.1, where both Sharpe ratio and maximum drawdown are in similar level. As α level increases, the rank of Sharpe ratio is lower. The performance measures are showed in figure 4.21.

Figure 4.21 Performance measures of CVaR model



From figure 4.22 shows the wealth and maximum drawdown. As α increase, the return and Sharpe ratio increased. The maximum drawdown decreased. There's no significant trend for standard deviation. Only when α=0.15, the strategy is not dominated.

4.6 Comparison of Models

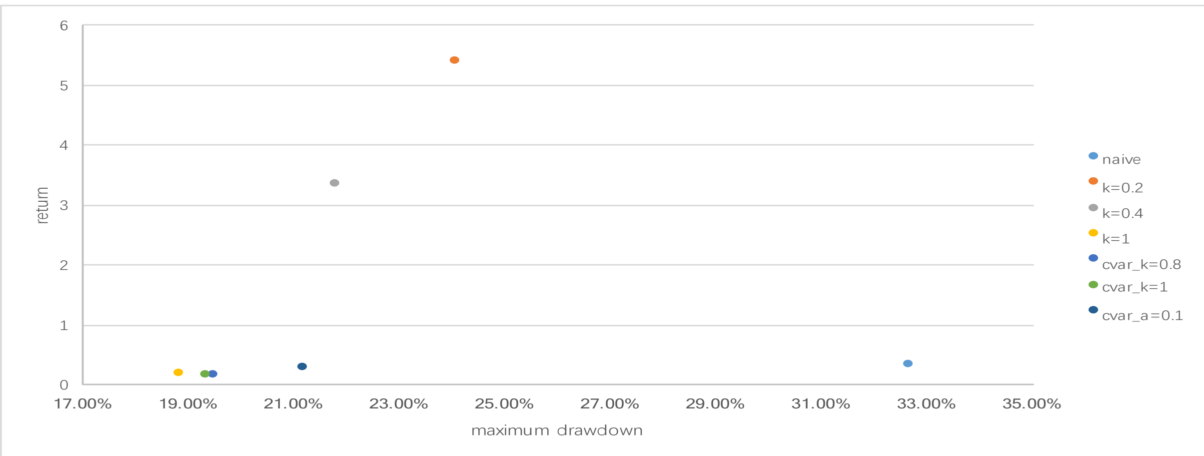
In this section we do the comparison of the naive strategy, Markowitz model and CVaR model. We've already found the optimal portfolios for these models. Now we make comparison of the optimal strategy that we gain. We divided the comparison to 2 parts. First we do the comparison for simple approach and then rolling window approach. We choose always 3 strategies from both Markowitz and CVaR model, one with the highest return, one with the highest Sharpe ratio and then one or two strategies from the 1st and 2nd rank of maximum drawdown (in the case that the previous two strategies is the same). Table 4.20 is the chosen strategies for simple approach.

Table 4.20 Comparison of models for simple approach

	Naive	Markowitz			CVaR		
Chosen parameter	-	k=0.2	k=0.4	k=1	k=0.8 α =0.15	k=1 α =0.15	α =0.1 k=0.2
Final wealth	4.05	105.07	68.62	3.14	3.31	3.23	5.01
Return	32.30%	539.93%	333.86%	17.73%	14.89%	14.95%	27.78%
Annual Return	5.74%	44.75%	33.96%	3.31%	2.80%	2.81%	5.00%
Standard deviation	16.70%	27.47%	23.90%	12.60%	12.97%	12.92%	13.42%
Maximum drawdown	32.66%	24.06%	21.80%	18.84%	19.48%	19.34%	21.20%
Sharpe ratio	3.48%	19.27%	17.09%	1.39%	0.62%	0.62%	2.95%

From table 4.20 we can see the highest return is when k=0.2 for Markowitz model and the second rank is when k=0.3. The rank for Sharpe ratio is the same. For maximum drawdown, the lowest level is k=1 for Markowitz model and 2nd rank is k=1 for CVaR model. Moreover, we can see the plot of maximum drawdown and return in figure 4.22.

Figure 4.22 Maximum drawdown and return of simple approach



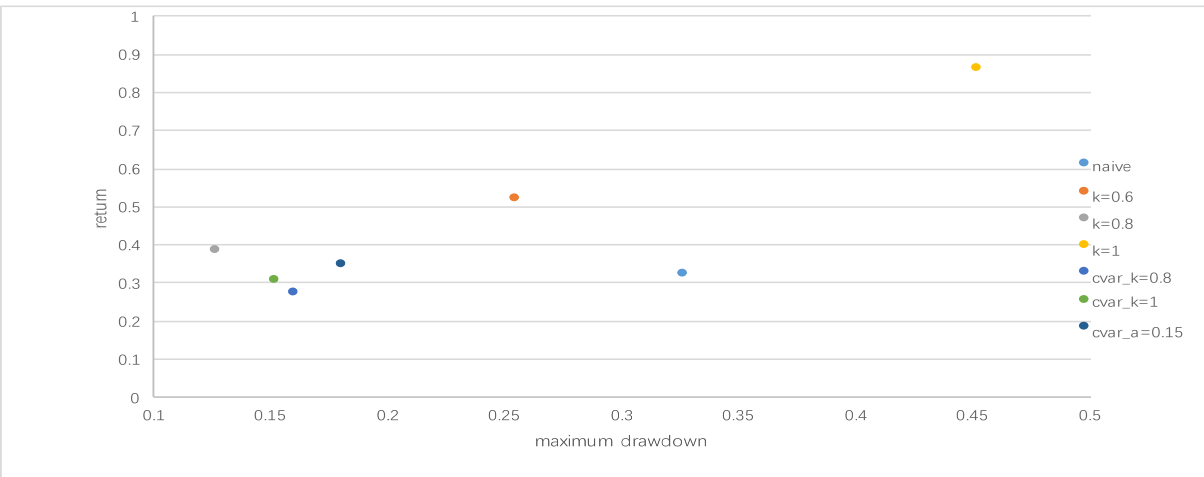
We want to choose lower maximum drawdown and higher return rate. Firstly we reject naive strategy because of the low return and high maximum drawdown. We can accept the situation when *k*=0.2 for Markowitz model, because it has 1st rank of wealth. Also we can accept when *k*=0.4 with second rank of level. When *k*=1, it has the best maximum drawdown. When *k*=0.8 foe CVaR model, the maximum drawdown and wealth is worse than k=1 for Markowitz model, so we can't accept the strategy. When *k*=1, *k*=0.4 and *k*=0.2 of Markowitz model are not denominated. These strategies can be accepted. We can do the same comparison for rolling window approach.

Table 4.21 Comparison of models for rolling window approach

	Naive	Markowitz			CVaR		
Chosen parameter	-	0.6	0.8	1	k=0.8 α =0.15	k=1 α =0.15	α =0.15 k=0.2
Final wealth	4.05	1.78	1.48	1.38	1.27	1.31	1.35
Return	32.30%	86.26%	52.04%	38.43%	27.14%	30.57%	34.57%
Annual Return	5.74%	13.19%	8.71%	6.69%	4.90%	5.46%	6.09%
Standard deviation	16.70%	24.63%	16.45%	12.09%	12.04%	11.96%	14.10%
Maximum drawdown	32.66%	45.15%	25.43%	12.65%	16.02%	15.17%	18.06%
Sharpe ratio	3.48%	7.14%	5.85%	5.09%	3.14%	3.77%	4.07%

For rolling window approach, the highest Sharpe ratio is when k=0.6 for Markowitz model. In this situation we also get highest maximum drawdown and standard deviation. The second rank is then *k*=0.8 for Markowitz model. The minimum maximum drawdown among all strategies is *k*=1 for Markowitz model. The lowest Table 4.22 is the plot of maximum drawdown for rolling window strategy.

Table 4.22 Comparison of maximum drawdown for rolling window strategy.



From table 4.22 we can see that point k=0.6, k=0.8 and k=1 for Markowitz model are acceptable strategies that are not denominated. When k=1, the strategy is with highest return rate and highest maximum drawdown. when k=0.6, the rank of return is the 2^{nd} and it has 3^{rd} level of maximum drawdown. We reject the naive strategy because it has worse maximum drawdown and return than k=0.6 for Markowitz model.

5. Conclusion

Portfolio optimization is a combination of various kinds of assets to spread the risk and get maximum return. In this thesis we analyze the portfolio optimization based on naive strategy, Markowitz model and CVaR model by using Matlab software. In chapter two, we did the basic introduction of the Matlab software. In chapter three, portfolio optimization models were introduced. Then, in chapter four we put the methodology into practice and analyze different strategies.

As we mentioned in chapter one, the goal of the thesis is to verify and compare the out-of-sample performance of naive, Markowitz and CVaR strategies. For naive strategie, we calculate the in-sample and out-of sample result. For Markowitz model, we compare the result of different levels of k . For CVaR model, we firstly set α level at constant and compare the result for different k levels, then we apply different α levels for calculation with constant k level. Finally we compare the optimal strategy among three models.

For Markowitz model, for simple approach as k level increases, the return has a decrease tendency. The standard deviation has a slightly decrease and maximum drawdown also has small decrease . For Sharpe ratio, it has a huge decrease. For rolling window strategy, k level from 0 to 0.6 is with the same tendency that the increase in k value causes a decrease in return, standard deviation and maximum drawdown. The Sharpe ratio increases.

For CVaR model, decrease in k level lead to increase in return, standard deviation, Sharpe ratio and maximum drawdown for most of chosen levels for both simple and rolling window strategies. For α levels from 0.005 to 0.05, the higher α level increases the return rate and Sharpe ratio and decrease the annual standard deviation and maximum drawdown.

From my point of view, I would prefer Markowitz when k=0.2 for simple approach. As I'm a risk averse investor and want to maximize return and minimize risk. Under this strategy the return is significantly higher than other strategies, the Sharpe ratio is also the highest. The maximum drawdown is in the middle level among all strategies. I'd like to take the risk and invest into this strategy.

Bibliography

1. ATSUSHI, I. and K. LUTZ. In-Sample or Out-of-Sample Tests of Predictability: Which One Should We Use? *Econometric Reviews*. 2005, Vol. 23, No. 4, pp. 371-402. ISSN 0747-4938.
2. BEUCHER, O. and M. WEEKSe. *Introduction to MATLAB & SIMULINK*. Hingham, Mass.: Infinity Science Press, 2008. ISBN 978-1-934015-04-9.
3. CAMPBELL, Sean. A review of backtesting and backtesting procedures. *The Journal of Risk*. 2006, Vol. 9, No. 2, pp. 1-17. ISSN 1465-1221.
4. DEMIGUEL, V., L. GARLAPPI and R. UPPAL. Optimal Versus Naive Diversification: How Inefficient is the 1/NPortfolio Strategy? *Review of Financial Studies*. 2007, Vol. 22, No. 5, pp. 1915-1953. ISSN 0893-9454.
5. HURLIN, C. and S. TOKPAVI. Backtesting value-at-risk accuracy: a new simple test. *The Journal of Risk*. 2007,Vol. 9, No. 2, pp. 19-37. ISSN 1465-1221.
6. KOPA, Miloš. Out-of-sample optimal risk parameter in mean- CVaR models. *In: 10th International Scientific Conference Financial management of Firms and Financial Institutions.* Ostrava: VŠB-TU Ostrava, 2015. pp. 544-549. ISBN 978-80-298-3865-6.
7. KRESTA, Aleš. *Financial Engineering in Matlab: Selected Approaches and Algorithms*. Ostrava: VŠB-TU Ostrava, 2015. ISBN 978-80-248-3702-4.
8. KRESTA, A. and K. ZELINKOVÁ. Backtesting of portfolio optimization with and without risk-free asset. *Central European Review of Economic Issues*. 2015, Vol. 18, No. 2, pp. 75-81. ISSN 1212-3951.
9. MARKOWITZ, Harry. Portfolio Selection. *The Journal of Finance*. 1952, Vol. 7, No. 1, pp. 77-91. ISSN 1540-6261.
10. ROCKFELLAR, T. and S. URYASEV. Conditional Value-at-Risk for General Loss Distributions. *Journal of Banking & Finance*. 2002, Vol. 26, No. 7, pp. 1443-1471. ISSN 0378-4266.
11. SALAHI, M., F. MEHRDOUST and F. PIRI. CVaR Robust Mean- CVaR Portfolio Optimization. *ISRN Applied Mathematics*. 2013, Vol. 2013, pp. 1-9. ISSN 2090-5564.
12. STEIN, Charles. Inadmissibility of the usual estimator for the variance of a normal distribution with unknown mean. *Annals of the Institute of Statistical Mathematics*. 1964, Vol. 16, No. 1, pp. 155-160. ISSN 0020-3157.

13. YU, X., H. SUN and G. CHEN. The Optimal Portfolio Model Based on Mean-CVaR. *Journal of Mathematical Finance*. 2011, Vol. 1, No. 3, pp. 132-134. ISSN 2162-2434.
14. ZMEŠKAL, Z., D. DLUHOŠOVÁ and T. TICHÝ. *Financial Models*. Ostrava: VŠB-TU Ostrava, 2004. ISBN 80-248-0754-8.

Electronic documents and others

15. https://finance.yahoo.com/

List of Abbreviations

CVaR	Conditional value at risk
Cov	Covariance
DD	Drawdown
$E\left(R_P\right)$	The expected return of portfolio
MDD	Maximum drawdown
$\boldsymbol{Q}$	Covariance matrix
$r_{i,t}$	Ex post returns
R_f	The risk free rate
U	Utility function
Var	Variance of portfolio
VaR	Value at risk
$w_{i,t}$	The weights of assets
$\rho(X)$	The mapping function from random variable to real numbers
σ_p	Standard deviation of portfolio

Declaration of Utilisation of Results from a Diploma Thesis

Herewith I declare that

— I am informed that Act No. 121/2000 Coll. – the Copyright Act, in particular, Section 35 – Utilisation of the Work as a Part of Civil and Religious Ceremonies, as a Part of School Performances and the Utilisation of a School Work – and Section 60 – School Work, fully applies to my diploma thesis;

— I take account of the VSB – Technical University of Ostrava (hereinafter as VSB-TUO) having the right to utilize the diploma thesis (under Section 35(3)) unprofitably and for own use ; I agree that the diploma thesis shall be archived in the electronic form in VSB- TUO's Central Library and one copy shall be kept by the supervisor of the diploma thesis. I agree that the bibliographic information about the diploma thesis shall be published in VSB-TUO's information system;

— It was agreed that, in case of VSB-TUO's interest, I shall enter into a license agreement with VSB-TUO, granting the authorization to utilize the work in the scope of Section 12(4) of the Copyright Act;

— It was agreed that I may utilize my work, the diploma thesis or provide a license to utilize it only with the consent of VSB-TUO, which is entitled, in such a case, to claim an adequate contribution from me to cover the cost expended by VSB-TUO for producing the work (up to its real amount).

Ostrava dated 18. 04. 2017

Yin Xu 徐寅

YIN XU

List of Annex

Annex A Matlab Programs

Program A1 Data Acquisition

```
%% Load data
load group_data
%%Plot the stock price
figure;
subplot(2,2,1)
plot(date,g1);
dateaxis('x');
xlabel('Date'); ylabel('Return');
legend(S1)
subplot(2,2,2)
plot(date,g2);
dateaxis('x');
xlabel('Date'); ylabel('Return');
legend(s2)
subplot(2,2,3)
plot(date,g3);
dateaxis('x');
xlabel('Date'); ylabel('Return');
legend(s3)
subplot(2,2,4)
plot(date,g4);
dateaxis('x');
xlabel('Date'); ylabel('Return');
legend(s4)
```

Program A2 Naive Strategy

```
%% load data
clear; clc;
load data;
%% devide data into 2 parts and calculate return
%1-311    in sample    %2006-2011
%312-end   out-of-sample %2012-2016
ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
%% 1/n strategy calculate return of portfolio
weights=repmat(1/43,43,1);
rp=ret*weights;
w=[1 ;cumprod(1+rp)];
%% calculation of results
insample=w(311)./w(1)-1;
outofsample=w(end)./w(312)-1;
r_in=power(1+insample,52/311)-1;
r_ot=power(1+outofsample,52/(572-311))-1
rf_rate=power(1.0284,1/52)-1;
in_std=std(rp(1:311))*sqrt(52)
os_std=std(rp(312:end))*sqrt(52)
max_in=perf_maxDD(w(1:311))
max_os=perf_maxDD(w(312:end))
SR_in=perf_Sharpe(rp(1:311),rf_rate); %Sharpe ratio
SR_os=perf_Sharpe(rp(312:end),rf_rate);
%% plot result
figure;
subplot(2,2,1);
plot(dates(1:310),rp(1:310));
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
title('In-sample Return')
subplot(2,2,2);
plot(dates(311:end),rp(311:end));
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
title('Out-of-sample Return')
subplot(2,2,3);
```

```
plot(date(1:311),w(1:311));
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
title('In-sample Wealth')
subplot(2,2,4);
plot(date(312:end),w(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
title('Out-of-sample Wealth')
```

Program A3   Generation of feasible set

```
%% load only in-sample data
clear; clc;
load insampledata;
%% devide data into 2 parts and calculate return
%1-311    in sample    %2006-2011
%312-end   out-of-sample %2012-2016
Returns=insampledata(2:end,:)./insampledata(1:end-1,:)-1;

%% the population parameters are estimated based on historical observations
eRs=mean(Returns)';
covariance=cov(Returns);
%% we generate the matrix consisting of possible weights' vectors
weights= generate_weights(size(eRs,1),0.25,0);
%fir each vector of weights we compute E(R) and standard dev.
for a=1:size(weights,1)
  eRp(a)=weights(a,:)*eRs;
smodch(a)=sqrt(weights(a,:)*covariance*weights(a,:)');
end;
%% we plot the figure of feasible set
plot(smodch.*100,eRp.*100,'.', 'MarkerEdgeColor',[0.8 0.8 0.8]);
xlabel('portfolio standard deviation (in %)');
ylabel('portfolio expected return (in %)');
%% find the portfolio with the minimum risk
% find the portfolio with the minimum risk
[ x, ERp_min, sd ]=portfolio_optimize( eRs, covariance, 0, 1, -1); % find the
expected return of the maximum-expected-return portfolio
ERp_max=max(eRs);
% find 99 inferior points
for j=1:1:101
R_jg=ERp_min+(j-1)/100*(ERp_max-ERp_min);
[x(:,j),ERp(j),sd(j)]=portfolio_optimize(eRs,...
    covariance,0,1,R_jg);
end
x(x<0.00001)=0;
%%
figure
```

```
plot(smodch.*100,eRp.*100,'.', 'MarkerEdgeColor',[0.8 0.8 0.8]);
xlabel('portfolio standard deviation (in %)');
ylabel('portfolio expected return (in %)');

hold on;
plot(sd*100,ERp*100,'Color',[0 0 0],'LineWidth',2);
```

Program A4 Markowitz model

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-311     in sample    %2006-2011
%312-end   out-of-sample %2012-2016
ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;
%% mean-var
k=0
weights1=poptimize_mvbkp(ret(1:311,:),k);
rp1=ret*weights1;
w1=[1 ;cumprod(1+rp1)];
insample1=w1(311)./w1(1)-1;
outofsample1=w1(end)./w1(312)-1;
r_in1=power(1+insample1,52/311)-1;
r_ot1=power(1+outofsample1,52/(572-311))-1;
in_std1=std(rp1(1:311))*sqrt(52);
os_std1=std(rp1(312:end))*sqrt(52);
max_in1=perf_maxDD(w1(1:311));
max_os1=perf_maxDD(w1(312:end));
SR_in1=perf_Sharpe(rp1(1:311),rf_rate); %Sharpe ratio
SR_os1=perf_Sharpe(rp1(312:end),rf_rate);
%%
k=0.2
weights2=poptimize_mvbkp(ret(1:311,:),k);
rp2=ret*weights2;
w2=[1 ;cumprod(1+rp2)];
insample2=w2(311)./w2(1)-1;
outofsample2=w2(end)./w2(312)-1;
r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
```

```
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);
%%
k=0.4
weights3=poptimize_mvbkp(ret(1:311,:),k);
rp3=ret*weights3;
w3=[1 ;cumprod(1+rp3)];
insample3=w3(311)./w3(1)-1;
outofsample3=w3(end)./w3(312)-1;
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);
k=0.6
weights4=poptimize_mvbkp(ret(1:311,:),k);
rp4=ret*weights4;
w4=[1 ;cumprod(1+rp4)];
insample4=w4(311)./w4(1)-1;
outofsample4=w4(end)./w4(312)-1;
r_in4=power(1+insample4,52/311)-1;
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);
k=0.8
weights5=poptimize_mvbkp(ret(1:311,:),k);
rp5=ret*weights5;
w5=[1 ;cumprod(1+rp5)];
insample5=w5(311)./w5(1)-1;
```

```
outofsample5=w5(end)./w5(312)-1;
r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);
k=1
weights6=poptimize_mvbkp(ret(1:311,:),k);
rp6=ret*weights6;
w6=[1 ;cumprod(1+rp6)];
insample6=w6(311)./w6(1)-1;
outofsample6=w6(end)./w6(312)-1;
r_in6=power(1+insample6,52/311)-1;
r_ot6=power(1+outofsample6,52/(572-311))-1;
in_std6=std(rp6(1:311))*sqrt(52);
os_std6=std(rp6(312:end))*sqrt(52);
max_in6=perf_maxDD(w6(1:311));
max_os6=perf_maxDD(w6(312:end));
SR_in6=perf_Sharpe(rp6(1:311),rf_rate); %Sharpe ratio
SR_os6=perf_Sharpe(rp6(312:end),rf_rate);
%%  Combine data
weights=[weights2 weights3 weights4 weights5 weights6];
rp=[rp2 rp3 rp4 rp5 rp6];
w=[w2 w3 w4 w5  w6];
insample=[insample2 insample3 insample4 insample5 insample6];
outofsample=[outofsample2 outofsample3 outofsample4 outofsample5 outofsample6];
r_in=[r_in2 r_in3 r_in4 r_in5 r_in6];
r_ot=[r_ot2 r_ot3 r_ot4 r_ot5 r_ot6];
in_std=[in_std2 in_std3 in_std4 in_std5 in_std6];
os_std=[os_std2 os_std3 os_std4 os_std5 os_std6];
max_in=[max_in2 max_in3 max_in4 max_in5 max_in6];
max_os=[max_os2 max_os3 max_os4 max_os5 max_os6];
SR_in=[SR_in2 SR_in3 SR_in4 SR_in5 SR_in6];
SR_os=[SR_os2 SR_os3 SR_os4 SR_os5 SR_os6];
%% plot results
```

```
figure;
subplot(3,2,1)
plot(dates,rp1);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0');
subplot(3,2,2)
plot(dates,rp2);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.2');
subplot(3,2,3)
plot(dates,rp3);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.4');
subplot(3,2,4)
plot(dates,rp4);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.6');
subplot(3,2,5)
plot(dates,rp5);
dateaxis('x'); xlabel('Date'); ylabel('Return'); title('k=0.8');
subplot(3,2,6)
plot(dates,rp6);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=1');
%%
figure;
plot(date,w);
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
legend('k=0','k=0.2','k=0.4','k=0.6','k=0.8','k=1');
hold on
plot([date(311),date(311)],[0,140],'k')
```

Program A5 Backtesting of Markowitz model

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-311     in sample    %2006-2011
%312-end   out-of-sample %2012-2016

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;
%% mean-var
k=0
for t=312:571
  weights1(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp1=sum(ret.*weights1,2);
w1=[1 ;cumprod(1+rp1)];
insample1=w1(311)./w1(1)-1;
outofsample1=w1(end)./w1(312)-1;
r_in1=power(1+insample1,52/311)-1;
r_ot1=power(1+outofsample1,52/(572-311))-1;
in_std1=std(rp1(1:311))*sqrt(52);
os_std1=std(rp1(312:end))*sqrt(52);
max_in1=perf_maxDD(w1(1:311));
max_os1=perf_maxDD(w1(312:end));
SR_in1=perf_Sharpe(rp1(1:311),rf_rate); %Sharpe ratio
SR_os1=perf_Sharpe(rp1(312:end),rf_rate);
%%
k=0.2
for t=312:571
  weights2(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp2=sum(ret.*weights2,2);
w2=[1 ;cumprod(1+rp2)];
insample2=w2(311)./w2(1)-1;
outofsample2=w2(end)./w2(312)-1;
```

```
r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);
%%

k=0.4
for t=312:571
  weights3(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp3=sum(ret.*weights3,2);
w3=[1 ;cumprod(1+rp3)];
insample3=w3(311)./w3(1)-1;
outofsample3=w3(end)./w3(312)-1;
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);



%%
k=0.6
for t=312:571
  weights4(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp4=sum(ret.*weights4,2);
w4=[1 ;cumprod(1+rp4)];
insample4=w4(311)./w4(1)-1;
outofsample4=w4(end)./w4(312)-1;
r_in4=power(1+insample4,52/311)-1;
```

```
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);

%%
k=0.8
for t=312:571
  weights5(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp5=sum(ret.*weights5,2);
w5=[1 ;cumprod(1+rp5)];
insample5=w5(311)./w5(1)-1;
outofsample5=w5(end)./w5(312)-1;
r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);

%%
k=1
for t=312:571
  weights6(t,:)=poptimize_mvbkp(ret(t-311:t-1,:),k);
end
rp6=sum(ret.*weights6,2);
w6=[1 ;cumprod(1+rp6)];
insample6=w6(311)./w6(1)-1;
outofsample6=w6(end)./w6(312)-1;
r_in6=power(1+insample6,52/311)-1;
r_ot6=power(1+outofsample6,52/(572-311))-1;
in_std6=std(rp6(1:311))*sqrt(52);
```

```
os_std6=std(rp6(312:end))*sqrt(52);
max_in6=perf_maxDD(w6(1:311));
max_os6=perf_maxDD(w6(312:end));
SR_in6=perf_Sharpe(rp6(1:311),rf_rate); %Sharpe ratio
SR_os6=perf_Sharpe(rp6(312:end),rf_rate);
%%  Combine data
weights=[weights1 weights2 weights3 weights4 weights5 weights6];
rp=[rp1 rp2 rp3 rp4 rp5 rp6];
w=[w1 w2 w3 w4 w5  w6];
insample=[insample1 insample2 insample3 insample4 insample5 insample6];
outofsample=[outofsample1 outofsample2 outofsample3 outofsample4 outofsample5
outofsample6];
r_in=[r_in1 r_in2 r_in3 r_in4 r_in5 r_in6];
r_ot=[r_ot1 r_ot2 r_ot3 r_ot4 r_ot5 r_ot6];
in_std=[in_std1 in_std2 in_std3 in_std4 in_std5 in_std6];
os_std=[os_std1 os_std2 os_std3 os_std4 os_std5 os_std6];
max_in=[max_in1 max_in2 max_in3 max_in4 max_in5 max_in6];
max_os=[max_os1 max_os2 max_os3 max_os4 max_os5 max_os6];
SR_in=[SR_in1 SR_in2 SR_in3 SR_in4 SR_in5 SR_in6];
SR_os=[SR_os1 SR_os2 SR_os3 SR_os4 SR_os5 SR_os6];
%% plot results
figure;
subplot(3,2,1)
plot(dates(312:end),rp1(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0');
subplot(3,2,2)
plot(dates(312:end),rp2(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.2');
subplot(3,2,3)
plot(dates(312:end),rp3(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.4');
subplot(3,2,4)
```

```
plot(dates(312:end),rp4(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.6');
subplot(3,2,5)
plot(dates(312:end),rp5(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.8');
subplot(3,2,6)
plot(dates(312:end),rp6(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=1');

%%
figure;
plot(date(312:end),w1(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
hold on;
plot(date(312:end),w2(312:end));
hold on;
plot(date(312:end),w3(312:end));
hold on;
plot(date(312:end),w4(312:end));
hold on;
plot(date(312:end),w5(312:end));
hold on ;
plot(date(312:end),w6(312:end));
legend('k=0','k=0.2','k=0.4','k=0.6','k=0.8','k=1');
```

Program A6 CVaR model for combination of k and α level

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-310     in sample    %2006-2011
%311-end   out-of-sample %2012-2016
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
r=ret(1:310,:)
%%
for a=1:99
   for b=1:6
alpha=a/100;
k=(b-1)/5;

weights(b,a,:)=poptimize_mcvarbkp(ret(1:310,:),k,alpha);
rp1=ret*squeeze(weights(b,a,:));
w1=[1 ;cumprod(1+rp1)];
insample1=w1(310)./w1(1)-1;
outofsample1=w1(end)./w1(311)-1;
r_ot(b,a)=power(1+outofsample1,52/(572-311))-1;
max_os(b,a)=perf_maxDD(w1(312:end));
SR_os(b,a)=perf_Sharpe(rp1(312:end),rf_rate);

   end
end
%%
figure;
contour(max_os,100);
figure;
contour(SR_os,100);
figure;
contour(r_ot,100);
```

Program A7 CVaR model of different *k* level

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-310     in sample    %2006-2011
%311-end   out-of-sample %2012-2016

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;
%% mean-CVAR
k=0.2
weights2=poptimize_mcvarbkp(ret(1:310,:),k,0.15);
rp2=ret*weights2;
w2=[1 ;cumprod(1+rp2)];
insample2=w2(310)./w2(1)-1;
outofsample2=w2(end)./w2(311)-1;
r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);

%%

k=0.4
weights3=poptimize_mcvarbkp(ret(1:310,:),k,0.15);
rp3=ret*weights3;
w3=[1 ;cumprod(1+rp3)];
insample3=w3(310)./w3(1)-1;
outofsample3=w3(end)./w3(311)-1;
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1;
```

```
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);
%%
k=0.6
weights4=poptimize_mcvarbkp(ret(1:310,:),k,0.15);
rp4=ret*weights4;
w4=[1 ;cumprod(1+rp4)];
insample4=w4(310)./w4(1)-1;
outofsample4=w4(end)./w4(311)-1;
r_in4=power(1+insample4,52/311)-1;
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);
%%
k=0.8
weights5=poptimize_mcvarbkp(ret(1:310,:),k,0.15);
rp5=ret*weights5;
w5=[1 ;cumprod(1+rp5)];
insample5=w5(310)./w5(1)-1;
outofsample5=w5(end)./w5(311)-1;
r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);
k=1
weights6=poptimize_mcvarbkp(ret(1:310,:),k,0.15);
```

```
rp6=ret*weights6;
w6=[1 ;cumprod(1+rp6)];
insample6=w6(310)./w6(1)-1;
outofsample6=w6(end)./w6(311)-1;
r_in6=power(1+insample6,52/311)-1;
r_ot6=power(1+outofsample6,52/(572-311))-1
in_std6=std(rp6(1:311))*sqrt(52);
os_std6=std(rp6(312:end))*sqrt(52);
max_in6=perf_maxDD(w6(1:311));
max_os6=perf_maxDD(w6(312:end));
SR_in6=perf_Sharpe(rp6(1:311),rf_rate); %Sharpe ratio
SR_os6=perf_Sharpe(rp6(312:end),rf_rate);
%%  Combine data
weights=[weights2 weights3 weights4 weights5 weights6]
rp=[rp2 rp3 rp4 rp5 rp6]
w=[w2 w3 w4 w5  w6]
insample=[insample2 insample3 insample4 insample5 insample6]
outofsample=[outofsample2 outofsample3 outofsample4 outofsample5 outofsample6]
r_in=[r_in2 r_in3 r_in4 r_in5 r_in6];
r_ot=[r_ot2 r_ot3 r_ot4 r_ot5 r_ot6];
in_std=[in_std2 in_std3 in_std4 in_std5 in_std6];
os_std=[os_std2 os_std3 os_std4 os_std5 os_std6];
max_in=[max_in2 max_in3 max_in4 max_in5 max_in6];
max_os=[max_os2 max_os3 max_os4 max_os5 max_os6];
SR_in=[SR_in2 SR_in3 SR_in4 SR_in5 SR_in6];
SR_os=[SR_os2 SR_os3 SR_os4 SR_os5 SR_os6];
%% plot results
figure;
subplot(3,2,1)
plot(dates,rp2);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.2');
subplot(3,2,2)
plot(dates,rp3);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.4');
```

```
subplot(3,2,3)
plot(dates,rp4);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.6');
subplot(3,2,4)
plot(dates,rp5);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.8');
subplot(3,2,5)
plot(dates,rp6);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=1');
%%
figure;
plot(date,w);
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
legend('k=0.2','k=0.4','k=0.6','k=0.8','k=1');
hold on
plot([date(311),date(311)],[0,8],'k')
```

Program A8 CVaR model of different *alpha* level

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-310     in sample    %2006-2011
%311-end   out-of-sample %2012-2016

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
r=ret(1:310,:)
%% mean-cvar
alpha=0.005
weights1=poptimize_mcvarbkp(ret(1:310,:),0.2,alpha);
rp1=ret*weights1;
w1=[1 ;cumprod(1+rp1)];
insample1=w1(310)./w1(1)-1;
outofsample1=w1(end)./w1(311)-1;
%%
alpha=0.01
weights2=poptimize_mcvarbkp(ret(1:310,:),0.2,alpha);
rp2=ret*weights2;
w2=[1 ;cumprod(1+rp2)];
insample2=w2(310)./w2(1)-1;
outofsample2=w2(end)./w2(311)-1;
%%

alpha=0.05
weights3=poptimize_mcvarbkp(ret(1:310,:),0.2,alpha);
rp3=ret*weights3;
w3=[1 ;cumprod(1+rp3)];
insample3=w3(310)./w3(1)-1;
outofsample3=w3(end)./w3(311)-1;
%%
alpha=0.1
weights4=poptimize_mcvarbkp(ret(1:310,:),0.2,alpha);
rp4=ret*weights4;
w4=[1 ;cumprod(1+rp4)];
```

```
insample4=w4(310)./w4(1)-1;
outofsample4=w4(end)./w4(311)-1;
%%
alpha=0.15
weights5=poptimize_mcvarbkp(ret(1:310,:),0.2,alpha);
rp5=ret*weights5;
w5=[1 ;cumprod(1+rp5)];
insample5=w5(310)./w5(1)-1;
outofsample5=w5(end)./w5(311)-1;

%% Combine data
weights=[weights1 weights2 weights3 weights4 weights5]
rp=[rp1 rp2 rp3 rp4 rp5]
w=[w1 w2 w3 w4 w5]
insample=[insample1 insample2 insample3 insample4 insample5]
outofsample=[outofsample1 outofsample2 outofsample3 outofsample4 outofsample5]

%% sr
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;
r_in1=power(1+insample1,52/311)-1;
r_ot1=power(1+outofsample1,52/(572-311))-1;
in_std1=std(rp1(1:311))*sqrt(52);
os_std1=std(rp1(312:end))*sqrt(52);
max_in1=perf_maxDD(w1(1:311));
max_os1=perf_maxDD(w1(312:end));
SR_in1=perf_Sharpe(rp1(1:311),rf_rate); %Sharpe ratio
SR_os1=perf_Sharpe(rp1(312:end),rf_rate);

r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);
```

```
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1;
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);

r_in4=power(1+insample4,52/311)-1;
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);

r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);
%%
r_in=[r_in1 r_in2 r_in3 r_in4 r_in5];
r_ot=[r_ot1 r_ot2 r_ot3 r_ot4 r_ot5];
in_std=[in_std1 in_std2 in_std3 in_std4 in_std5];
os_std=[os_std1 os_std2 os_std3 os_std4 os_std5];
max_in=[max_in1 max_in2 max_in3 max_in4 max_in5];
max_os=[max_os1 max_os2 max_os3 max_os4 max_os5];
SR_in=[SR_in1 SR_in2 SR_in3 SR_in4 SR_in5];
SR_os=[SR_os1 SR_os2 SR_os3 SR_os4 SR_os5];


%%
%% plot results
```

```
figure;
subplot(3,2,1)
plot(dates,rp1);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.005');
subplot(3,2,2)
plot(dates,rp2);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.01');
subplot(3,2,3);
plot(dates,rp3);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.05');
subplot(3,2,4)
plot(dates,rp4);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.1');
subplot(3,2,5)
plot(dates,rp5);
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.15');

%% insample result
alpha=[0.005 0.01 0.05 0.1 0.15]
%%
figure;
plot(date,w);
dateaxis('x');
xlabel('Date'); ylabel('Wealth');
legend('alpha=0.005','alpha=0.01','alpha=0.05','alpha=0.1','alpha=0.15');
hold on
plot([date(311),date(311)],[0,10],'k')
```

Program A9 Backtesting of CVaR model of different k

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-311      in sample    %2006-2011
%312-end    out-of-sample %2012-2016

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;

%%
k=0.2
for t=312:571
  weights2(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),k,0.15);
end
rp2=sum(ret.*weights2,2);
w2=[1 ;cumprod(1+rp2)];
insample2=w2(311)./w2(1)-1;
outofsample2=w2(end)./w2(312)-1;
r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);
%%

k=0.4
for t=312:571
    weights3(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),k,0.15);
end
rp3=sum(ret.*weights3,2);
w3=[1 ;cumprod(1+rp3)];
```

```
insample3=w3(311)./w3(1)-1;
outofsample3=w3(end)./w3(312)-1;
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);



%%
k=0.6
for t=312:571
    weights4(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),k,0.15);
end
rp4=sum(ret.*weights4,2);
w4=[1 ;cumprod(1+rp4)];
insample4=w4(311)./w4(1)-1;
outofsample4=w4(end)./w4(312)-1;
r_in4=power(1+insample4,52/311)-1;
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);

%%
k=0.8
for t=312:571
    weights5(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),k,0.15);
end
rp5=sum(ret.*weights5,2);
w5=[1 ;cumprod(1+rp5)];
insample5=w5(311)./w5(1)-1;
```

```
outofsample5=w5(end)./w5(312)-1;
r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);

%%
k=1
for t=312:571
  weights6(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),k,0.15);
end
rp6=sum(ret.*weights6,2);
w6=[1 ;cumprod(1+rp6)];
insample6=w6(311)./w6(1)-1;
outofsample6=w6(end)./w6(312)-1;
r_in6=power(1+insample6,52/311)-1;
r_ot6=power(1+outofsample6,52/(572-311))-1;
in_std6=std(rp6(1:311))*sqrt(52);
os_std6=std(rp6(312:end))*sqrt(52);
max_in6=perf_maxDD(w6(1:311));
max_os6=perf_maxDD(w6(312:end));
SR_in6=perf_Sharpe(rp6(1:311),rf_rate); %Sharpe ratio
SR_os6=perf_Sharpe(rp6(312:end),rf_rate);
%%  Combine data
weights=[weights2 weights3 weights4 weights5 weights6];
rp=[rp2 rp3 rp4 rp5 rp6];
w=[w2 w3 w4 w5  w6];
insample=[insample2 insample3 insample4 insample5 insample6];
outofsample=[outofsample2 outofsample3 outofsample4 outofsample5 outofsample6];
r_in=[r_in2 r_in3 r_in4 r_in5 r_in6];
r_ot=[r_ot2 r_ot3 r_ot4 r_ot5 r_ot6];
in_std=[in_std2 in_std3 in_std4 in_std5 in_std6];
os_std=[os_std2 os_std3 os_std4 os_std5 os_std6];
max_in=[max_in2 max_in3 max_in4 max_in5 max_in6];
```

```
max_os=[max_os2 max_os3 max_os4 max_os5 max_os6];
SR_in=[SR_in2 SR_in3 SR_in4 SR_in5 SR_in6];
SR_os=[SR_os2 SR_os3 SR_os4 SR_os5 SR_os6];
%% plot results
figure;
subplot(3,2,1)
plot(dates(312:end),rp2(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.2');
subplot(3,2,2)
plot(dates(312:end),rp3(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.4');
subplot(3,2,3)
plot(dates(312:end),rp4(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.6');
subplot(3,2,4)
plot(dates(312:end),rp5(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=0.8');
subplot(3,2,5)
plot(dates(312:end),rp6(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('k=1');

%%
figure;

plot(date(312:end),w2(312:end));
hold on;
plot(date(312:end),w3(312:end));
hold on;
```

```
plot(date(312:end),w4(312:end));
hold on;
plot(date(312:end),w5(312:end));
hold on ;
plot(date(312:end),w6(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
legend('k=0.2','k=0.4','k=0.6','k=0.8','k=1');
```

Program A10 Backtesting of CVaR model of different α

```
%% load data
clear;
clc;
load data;
%% devide data into 2 parts and calculate return
%1-311     in sample    %2006-2011
%312-end   out-of-sample %2012-2016

ret=stocks(2:end,:)./stocks(1:end-1,:)-1;
rf_rate=power(1.0284,1/52)-1; %weekly return of risk free rate on 10 years
government bonds;

%%
alpha=0.01
for t=312:571
  weights2(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),0.2,alpha);
end
rp2=sum(ret.*weights2,2);
w2=[1 ;cumprod(1+rp2)];
insample2=w2(311)./w2(1)-1;
outofsample2=w2(end)./w2(312)-1;
r_in2=power(1+insample2,52/311)-1;
r_ot2=power(1+outofsample2,52/(572-311))-1;
in_std2=std(rp2(1:311))*sqrt(52);
os_std2=std(rp2(312:end))*sqrt(52);
max_in2=perf_maxDD(w2(1:311));
max_os2=perf_maxDD(w2(312:end));
SR_in2=perf_Sharpe(rp2(1:311),rf_rate); %Sharpe ratio
SR_os2=perf_Sharpe(rp2(312:end),rf_rate);
%%

alpha=0.05
for t=312:571
    weights3(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),0.2,alpha);
end
rp3=sum(ret.*weights3,2);
w3=[1 ;cumprod(1+rp3)];
insample3=w3(311)./w3(1)-1;
outofsample3=w3(end)./w3(312)-1;
r_in3=power(1+insample3,52/311)-1;
r_ot3=power(1+outofsample3,52/(572-311))-1
in_std3=std(rp3(1:311))*sqrt(52);
os_std3=std(rp3(312:end))*sqrt(52);
max_in3=perf_maxDD(w3(1:311));
max_os3=perf_maxDD(w3(312:end));
SR_in3=perf_Sharpe(rp3(1:311),rf_rate); %Sharpe ratio
SR_os3=perf_Sharpe(rp3(312:end),rf_rate);
```

```
%%
alpha=0.1
for t=312:571
   weights4(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),0.2,alpha);
end
rp4=sum(ret.*weights4,2);
w4=[1 ;cumprod(1+rp4)];
insample4=w4(311)./w4(1)-1;
outofsample4=w4(end)./w4(312)-1;
r_in4=power(1+insample4,52/311)-1;
r_ot4=power(1+outofsample4,52/(572-311))-1;
in_std4=std(rp4(1:311))*sqrt(52);
os_std4=std(rp4(312:end))*sqrt(52);
max_in4=perf_maxDD(w4(1:311));
max_os4=perf_maxDD(w4(312:end));
SR_in4=perf_Sharpe(rp4(1:311),rf_rate); %Sharpe ratio
SR_os4=perf_Sharpe(rp4(312:end),rf_rate);

%%
alpha=0.15
for t=312:571
   weights5(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),0.2,alpha);
end
rp5=sum(ret.*weights5,2);
w5=[1 ;cumprod(1+rp5)];
insample5=w5(311)./w5(1)-1;
outofsample5=w5(end)./w5(312)-1;
r_in5=power(1+insample5,52/311)-1;
r_ot5=power(1+outofsample5,52/(572-311))-1;
in_std5=std(rp5(1:311))*sqrt(52);
os_std5=std(rp5(312:end))*sqrt(52);
max_in5=perf_maxDD(w5(1:311));
max_os5=perf_maxDD(w5(312:end));
SR_in5=perf_Sharpe(rp5(1:311),rf_rate); %Sharpe ratio
SR_os5=perf_Sharpe(rp5(312:end),rf_rate);

%%
alpha=0.005
for t=312:571
  weights6(t,:)=poptimize_mcvarbkp(ret(t-311:t-1,:),0.2,alpha);
end
rp6=sum(ret.*weights6,2);
w6=[1 ;cumprod(1+rp6)];
insample6=w6(311)./w6(1)-1;
outofsample6=w6(end)./w6(312)-1;
r_in6=power(1+insample6,52/311)-1;
```

```
r_ot6=power(1+outofsample6,52/(572-311))-1;
in_std6=std(rp6(1:311))*sqrt(52);
os_std6=std(rp6(312:end))*sqrt(52);
max_in6=perf_maxDD(w6(1:311));
max_os6=perf_maxDD(w6(312:end));
SR_in6=perf_Sharpe(rp6(1:311),rf_rate); %Sharpe ratio
SR_os6=perf_Sharpe(rp6(312:end),rf_rate);
%% Combine data
weights=[ weights6 weights2 weights3 weights4 weights5];
rp=[rp6 rp2 rp3 rp4 rp5];
w=[w6 w2 w3 w4 w5];
insample=[insample2 insample3 insample4 insample5 insample6];
outofsample=[outofsample6 outofsample2 outofsample3 outofsample4 outofsample5 ];
r_in=[r_in6 r_in2 r_in3 r_in4 r_in5 ];
r_ot=[r_ot6 r_ot2 r_ot3 r_ot4 r_ot5 ];
in_std=[ in_std2 in_std3 in_std4 in_std5 in_std6];
os_std=[os_std6 os_std2 os_std3 os_std4 os_std5 ];
max_in=[max_in2 max_in3 max_in4 max_in5 max_in6];
max_os=[ max_os6 max_os2 max_os3 max_os4 max_os5];
SR_in=[SR_in6 SR_in2 SR_in3 SR_in4 SR_in5];
SR_os=[SR_os6 SR_os2 SR_os3 SR_os4 SR_os5];
%% plot results
%% plot results
figure;
subplot(3,2,1)
plot(dates(312:end),rp6(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.005');
subplot(3,2,2)
plot(dates(312:end),rp2(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.01');
subplot(3,2,3);
plot(dates(312:end),rp3(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.05');
subplot(3,2,4)
plot(dates(312:end),rp4(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
title('alpha=0.1');
subplot(3,2,5)
plot(dates(312:end),rp5(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
```

```
title('alpha=0.15');
%%
figure;
plot(date(312:end),w6(312:end));
dateaxis('x');
xlabel('Date'); ylabel('Return');
hold on;
plot(date(312:end),w2(312:end));
hold on;
plot(date(312:end),w3(312:end));
hold on;
plot(date(312:end),w4(312:end));
hold on;
plot(date(312:end),w5(312:end));
hold on ;
plot(date(312:end),w6(312:end));
legend('alpha=0.005','alpha=0.01','alpha=0.05','alpha=0.1','alpha=0.15');
```

Program A11 Function of portfolio optimization of Markowitz model

```
function [v]=poptimize_mvbkp(R,k)
n=size(R,2);
r=mean(R);
Q=cov(R);
A=[]; b=[];          %  A*x<=b
Aeq=ones(1,n); beq=1; %Aeq*x=beq
lb=repmat(0,n,1);     %lb<=x<=ub
ub=repmat(1,n,1);     %lb<=x<=ub
fun=@(x)(100000.*(x'*Q*x).*k-100000.*(r*x).*(1-k)); %optimaliza?n? funkce
options=optimoptions(@fmincon,'Algorithm','active-set','display','off','TolX',
5e-4);
[v,U,~]=fmincon(fun,repmat(1/n,n,1),A,b,Aeq,beq,lb,ub,[],options);
end
```

Program A12 Function of portfolio optimization of CVaR model

```
function [v]=poptimize_mcvarbkp(R,k,alfa)
n=size(R,2);
A=[]; b=[];          %  A*x<=b
Aeq=ones(1,n); beq=1; %Aeq*x=beq
lb=repmat(0,n,1);     %lb<=x<=ub
```

```
ub=repmat(1,n,1);     %lb<=x<=ub
fun=@(x)100000.*CVaR(R*x,alfa).*k-100000.*mean(R*x).*(1-k); %optimaliza?n?
funkce
options=optimoptions(@fmincon,'Algorithm','active-set','display','off','TolX',
5e-4);
[v,U,~]=fmincon(fun,repmat(1/n,n,1),A,b,Aeq,beq,lb,ub,[],options);
end
```

Program A13 Function of CVaR model

```
function [CVaR]=CVaR(vynosy,alfa)
vynosy=sort(vynosy);
n=length(vynosy);
index=floor(n*alfa);
CVaR=-(1/alfa).*(sum(vynosy(1:index-1,:),1)./n+(alfa-(index-1)./n).*vynosy(ind
ex,:));
end
```